EURASIP Book Series on Signal Processing and Communications

Genomic Signal Processing and Statistics

Edited by: Edward R. Dougherty, Ilya Shmulevich, Jie Chen, and Z. Jane Wang

Genomic Signal Processing and Statistics

EURASIP Book Series on Signal Processing and Communications
Editor-in-Chief: K. J. Ray Liu
Editorial Board: Zhi Ding, Moncef Gabbouj, Peter Grant, Ferran Marqués, Marc Moonen, Hideaki Sakai, Giovanni Sicuranza, Bob Stewart, and Sergios Theodoridis

Hindawi Publishing Corporation
410 Park Avenue, 15th Floor, #287 pmb, New York, NY 10022, USA
Nasr City Free Zone, Cairo 11816, Egypt
Fax: +1-866-HINDAWI (USA toll-free)

ISBN 977-5945-07-0

EURASIP Book Series on Signal Processing and Communications, Volume 2

Genomic Signal Processing and Statistics

Edited by: Edward R. Dougherty, Ilya Shmulevich, Jie Chen, and Z. Jane Wang

Hindawi Publishing Corporation
http://www.hindawi.com

Contents

Genomic signal processing: perspectives

Edward R. Dougherty, Ilya Shmulevich, Jie Chen, and Z. Jane Wang

No single agreed-upon definition seems to exist for the term *bioinformatics*, which has been used to mean a variety of things, ranging in scope and focus. To cite but a few examples from textbooks, Lodish et al. state that "bioinformatics is the rapidly developing area of computer science devoted to collecting, organizing, and analyzing DNA and protein sequences" [1]. A more general and encompassing definition, given by Brown, is that bioinformatics is "the use of computer methods in studies of genomes" [2]. More general still, "bioinformatics is the science of refining biological information into biological knowledge using computers" [3]. Kohane et al. observe that the "breadth of this commonly used definition of bioinformatics risks relegating it to the dustbin of labels too general to be useful" and advocate being more specific about the particular bioinformatics techniques employed [4].

Genomic signal processing (GSP) is the engineering discipline that studies the processing of genomic signals, by which we mean the measurable events, principally the production of mRNA and protein, that are carried out by the genome. Based upon current technology, GSP primarily deals with extracting information from gene expression measurements. The analysis, processing, and use of genomic signals for gaining biological knowledge constitute the domain of GSP. The aim of GSP is to integrate the theory and methods of signal processing with the global understanding of functional genomics, with special emphasis on genomic regulation [5]. Hence, GSP encompasses various methodologies concerning expression profiles: detection, prediction, classification, control, and statistical and dynamical modeling of gene networks. GSP is a fundamental discipline that brings to genomics the structural model-based analysis and synthesis that form the basis of mathematically rigorous engineering.

Recent methods facilitate large-scale surveys of gene expression in which transcript levels can be determined for thousands of genes simultaneously. In particular, expression microarrays result from a complex biochemical-optical system incorporating robotic spotting and computer image formation and analysis [6, 7, 8, 9, 10]. Since transcription control is accomplished by a method that interprets a variety of inputs, we require analytical tools for the expression profile data

that can detect the types of multivariate influences on decision making produced by complex genetic networks. Put more generally, signals generated by the genome must be processed to characterize their regulatory effects and their relationship to changes at both the genotypic and phenotypic levels. Application is generally directed towards tissue classification and the discovery of signaling pathways.

Because transcriptional control is accomplished by a complex method that interprets a variety of inputs, the development of analytical tools that detect multivariate influences on decision making present in complex genetic networks is essential. To carry out such an analysis, one needs appropriate analytical methodologies. Perhaps the most salient aspect of GSP is that it is an engineering discipline, having strong roots in signals and systems theory. In GSP, the point of departure is that the living cell is a system in which many interacting components work together to give rise to execution of normal cellular functions, complex behavior, and interaction with the environment, including other cells. In such systems, the "whole" is often more than the "sum of its parts," frequently referred to as emergent or complex behavior. The collective behavior of all relevant components in a cell, such as genes and their products, follows a similar paradigm, but gives rise to much richer behavior, that is characteristic of living systems. To gain insight into the behavior of such systems, a systems-wide approach must be taken. This requires us to produce a model of the components and their interactions and apply mathematical, statistical, or simulation tools to understand its behavior, especially as it relates to experimental data.

In this introductory chapter, we comment on four major areas of GSP research: signal extraction, phenotype classification, clustering, and gene regulatory networks. We then provide brief descriptions of each of the contributed chapters.

Signal extraction

Since a cell's specific functionality is largely determined by the genes it is expressing, it is logical that transcription, the first step in the process of converting the genetic information stored in an organism's genome into protein, would be highly regulated by the control network that coordinates and directs cellular activity. A primary means for regulating cellular activity is the control of protein production via the amounts of mRNA expressed by individual genes. The tools to build an understanding of genomic regulation of expression will involve the characterization of these expression levels. Microarray technology, both complementary DNA (cDNA) and oligonucleotide, provides a powerful analytic tool for genetic research. Since our concern is GSP, not microarray technology, we confine our brief discussion to cDNA microarrays.

Complementary DNA microarray technology combines robotic spotting of small amounts of individual, pure nucleic acid species on a glass surface, hybridization to this array with multiple fluorescently labeled nucleic acids, and detection and quantitation of the resulting fluor-tagged hybrids with a scanning confocal microscope. cDNA microarrays are prepared by printing thousands of cDNAs in an array format on glass microscope slides, which provide gene-specific hybridization targets. Distinct mRNA samples can be labeled with different fluors and then

cohybridized onto each arrayed gene. Ratios or direct intensity measurements of gene-expression levels between the samples can be used to detect meaningfully different expression levels between the samples for a given gene, the better choice depending on the sources of variation [11].

A typical glass-substrate and fluorescent-based cDNA microarray detection system is based on a scanning confocal microscope, where two monochrome images are obtained from laser excitations at two different wavelengths. Monochrome images of the fluorescent intensity for each fluor are combined by placing each image in the appropriate color channel of an RGB image. In this composite image, one can visualize the differential expression of genes in the two cell types: the test sample typically placed in the red channel, the reference sample in the green channel. Intense red fluorescence at a spot indicates a high level of expression of that gene in the test sample with little expression in the reference sample. Conversely, intense green fluorescence at a spot indicates relatively low expression of that gene in the test sample compared to the reference. When both test and reference samples express a gene at similar levels, the observed array spot is yellow. Assuming that specific DNA products from two samples have an equal probability of hybridizing to the specific target, the fluorescent intensity measurement is a function of the amount of specific RNA available within each sample, provided samples are wellmixed and there is sufficiently abundant cDNA deposited at each target location.

When using cDNA microarrays, the signal must be extracted from the background. This requires image processing to extract signals, variability analysis, and measurement quality assessment [12]. The objective of the microarray image analysis is to extract probe intensities or ratios at each cDNA target location and then cross-link printed clone information so that biologists can easily interpret the outcomes and high-level analysis can be performed. A microarray image is first segmented into individual cDNA targets, either by manual interaction or by an automated algorithm. For each target, the surrounding background fluorescent intensity is estimated, along with the exact target location, fluorescent intensity, and expression ratios.

In a microarray experiment, there are many sources of variation. Some types of variation, such as differences of gene expressions, may be highly informative as they may be of biological origin. Other types of variation, however, may be undesirable and can confound subsequent analysis, leading to wrong conclusions. In particular, there are certain systematic sources of variation, usually owing to a particular microarray technology, that should be corrected prior to further analysis. The process of removing such systematic variability is called normalization. There may be a number of reasons for normalizing microarray data. For example, there may be a systematic difference in quantities of starting RNA, resulting in one sample being consistently overrepresented. There may also be differences in labeling or detection efficiencies between the fluorescent dyes (e.g., Cy3, Cy5), again leading to systematic overexpression of one of the samples. Thus, in order to make meaningful biological comparisons, the measured intensities must be properly adjusted to counteract such systematic differences.

A major barrier to an effective understanding of variation is the large number of sources of variance inherent in microarray measurements. In many statistical analysis publications, the measured gene expression data are assumed to have multiple noise sources: noise due to sample preparation, labeling, hybridization, background fluorescence, different arrays, fluorescent dyes, and different printing locations. In attempting to quantify the noise level in a set of experiments, some studies employ ANOVA models in which the log-transformed gene expression signal is represented by true signal plus an additive noise [13, 14]. Other proposed models for expression signals include mixture models for gene effect [15], multiplicative model (not logarithm-transformed) [16, 17], ratio-distribution model [12, 18], binary model [19], rank-based models not sensitive to noise distributions [20], replicates using mixed models [21], quantitative noise analysis [22, 23], and design of reverse dye microarrays [24]. In addition to the many studies on noise estimation in microarrays, there is a large literature dealing with methods to isolate and eliminate the noise component from the measured signal. These studies suffer from the daunting complexity and inhomogeneity of the noise.

Classification

Pattern classification plays an important role in genomic signal analysis. For instance, cDNA microarrays can provide expression measurements for thousands of genes at once, and a key goal is to perform classification via different expression patterns. This requires designing a classifier that takes a vector of gene expression levels as input, and outputs a class label that predicts the class containing the input vector. Classification can be between different kinds of cancer, different stages of tumor development, or a host of such differences. Early cancer studies include leukemias [25] and breast cancer [26, 27]. Classifiers are designed from a sample of expression vectors by assessing expression levels from RNA obtained from the different tissues with microarrays, determining genes whose expression levels can be used as classifier variables, and then applying some rule to design the classifier from the sample microarray data.

An expression-based classifier provides a list of genes whose product abundance is indicative of important differences in a cell state, such as healthy or diseased, or one particular type of cancer or another. Among such informative genes are those whose products play a role in the initiation, progression, or maintenance of the disease. Two central goals of molecular analysis of disease are to use such information to directly diagnose the presence or type of disease and to produce therapies based on the mitigation of the aberrant function of gene products whose activities are central to the pathology of a disease. Mitigation would be accomplished either by the use of drugs already known to act on these gene products or by developing new drugs targeting these gene products.

Three critical statistical issues arise for expression-based classification [28]. First, given a set of variables, how does one design a classifier from the sample data that provides good classification over the general population? Second, how does one estimate the error of a designed classifier when data is limited? Third,

given a large set of potential variables, such as the large number of expression level determinations provided by microarrays, how does one select a set of variables as the input vector to the classifier? The difficulty of successfully accomplishing these tasks is severely exacerbated by the fact that small samples are ubiquitous in studies employing expression microarrays, meaning that the potential number of variables (gene expressions) is huge in comparison to the sample size (number of microarrays) [29]. As with most studies, due to cost and patient availability, this investigation will be in the small-sample category. Three points must be taken into consideration: (1) to avoid overfitting, simple classifiers should be employed [28, 30, 31]; (2) again to avoid overfitting, small feature sets are required [32, 33, 34, 35]; and (3) because samples are small and error estimation must be performed using the training data, the choice of error estimation rule is critical [36, 37], with feature-set ranking being of particular importance in gene discovery [38].

The problem of small-sample error estimation is particularly troublesome. An error estimator may be unbiased but have a large variance, and therefore, often be low. This can produce a large number of feature sets and classifiers with low error estimates. In the other direction, a small sample size enhances the possibility that a designed classifier will perform worse than the optimal classifier. Combined with a high error estimate, the result will be that many potentially good diagnostic gene sets will be pessimistically evaluated.

Not only is it important to base classifiers on small numbers of genes from a statistical perspective, there are compelling biological reasons for small classifier sets. As previously noted, correction of an aberrant function would be accomplished by the use of drugs. Sufficient information must be vested in gene sets small enough to serve as either convenient diagnostic panels or as candidates for the very expensive and time-consuming analysis required to determine if they could serve as useful targets for therapy. Small gene sets are necessary to allow construction of a practical immunohistochemical diagnostic panel. In sum, it is important to develop classification algorithms specifically tailored for small samples.

Clustering

A classifier takes a single data point (expression vector) and outputs a class label (phenotype); a cluster operator takes a set of data points (expression vectors) and partitions the points into clusters (subsets). Clustering has become a popular data-analysis technique in genomic studies using gene-expression microarrays [39, 40]. Time-series clustering groups together genes whose expression levels exhibit similar behavior through time. Similarity indicates possible coregulation. Another way to use expression data is to take expression profiles over various tissue samples, and then cluster these samples based on the expression levels for each sample, the motivation being the potential to discriminate pathologies based on their differential patterns of gene expression. A host of clustering algorithms has been proposed in the literature and many of these have been applied to genomic data: k-means, fuzzy c-means, self-organizing maps [41, 42, 43], hierarchical clustering, and model-based clustering [44, 45].

Many validation techniques have been proposed for evaluating clustering results. These are generally based on the degree to which clusters derived from a set of sample data satisfy certain heuristic criteria. This is significantly different than classification, where the error of a classifier is given by the probability of an erroneous decision. Validation methods can be roughly divided into two categories (although this categorization can certainly be made finer)—*internal* and *external*.

Internal validation methods evaluate the clusters based solely on the data, without external information. Typically, a heuristic measure is defined to indicate the goodness of the clustering. It is important to keep in mind that the measure only applies to the data at hand, and therefore is not predictive of the worth of a clustering algorithm—even with respect to the measure itself. Since these kinds of measures do not possess predictive capability, it appears difficult to assess their worth—even what it means to be "worthy." But there have been simulation studies to observe how they behave [46].

External validation methods evaluate a clustering algorithm by comparing the resulting clusters with prespecified information [47]. Agreement between the heuristic and algorithm-based partitions indicates algorithm accuracy. It also indicates that the scientific understanding behind the heuristic partition is being reflected in the measurements, thereby providing supporting evidence for the measurement process.

With model-based clustering, a Bayesian approach can be taken to determine the best number of clusters. Two models can be compared relative to the sample data by a *Bayes factor* [48, 49].

To recognize the fundamental difference between clustering and classification, we note two key characteristics of classification: (1) classifier error can be estimated under the assumption that the sample data arise from an underlying feature-label distribution; and (2) given a family of classifiers, sample data can be used to learn the optimal classifier in the family. Once designed, the classifier represents a mathematical model that provides a decision mechanism relative to real-world measurements. The model represents scientific knowledge to the extent that it has predictive capability. The purpose of testing (error estimation) is quantifying the worth of the model. Clustering has generally lacked both fundamental characteristics of classification. In particular, lacking inference in the context of a probability model, it has remained essentially a subjective visualization tool. Jain et al. wrote, "Clustering is a subjective process; the same set of data items often needs to be partitioned differently for different applications. This subjectivity makes the process of clustering difficult" [50]. Duda et al. stated the matter radically, "The answer to whether or not it is possible in principle to learn anything from unlabeled data depends upon the assumptions one is willing to accept—theorems cannot be proved without premises" [51]. These criticisms raise the question as to whether clustering can be used for scientific knowledge. This issue has been raised specifically in the context of gene-expression microarrays by Kerr and Churchill when they wrote, "A great deal of effort has gone into identifying the best clustering techniques for microarray data. However, another question that is at least

as important has received less attention; how does one make statistical inferences based on the results of clustering?" [52]. Indeed, how is one going to judge the relative worth of clustering algorithms unless it is based on their inference capabilities?

For clustering to have a sound scientific basis, error estimation must be addressed in the context of an appropriate probabilistic model. Ipso facto, since a clustering algorithm partitions a set of data points, error estimation for clustering must assume that clusters resulting from a cluster algorithm can be compared to the correct clusters for the data set in the context of a probability distribution, thereby providing an error measure. The key to a general probabilistic theory of clustering, including both error estimation and learning, is to recognize that classification theory is based on operators on random variables, and that the theory of clustering needs to be based on operators on random points sets [53]. Once clustering has been placed into a probabilistic context, proposed clustering algorithms can be rigorously evaluated as estimators, rules can be developed from designing clustering algorithms from data (analogous to the design of classifiers via classification rules), and these rules can be evaluated based on the kinds of criteria used for classification rules, such as consistency, approximation, and sample size.

Gene regulatory networks

Cellular control and its failure in disease result from multivariate activity among cohorts of genes. Thus, for therapeutic purposes, it is important to model this multivariate interaction. In the literature, two somewhat distinct approaches have been taken to carry out this modeling. The first approach is based on constructing detailed biochemical network models for particular cellular reactions of interest and makes use of ordinary differential equations, partial differential equations, and their variants [54]. While this method yields insights into the details of individual reaction pathways, it is not clear how the information obtained can be used to design a therapeutic regimen for a complex disease like cancer, which simultaneously involves many genes and many signaling pathways. A major problem for fine-scale modeling is its large data requirement. A second approach involves building coarse models of genetic interaction using the limited amount of microarray gene expression data that is usually available. Paradigms that have been considered in this context include directed graphs, Bayesian networks, Boolean networks, generalized logical networks, and probabilistic gene regulatory networks (PGRNs), which include the special case of probabilistic Boolean networks (PBNs).

Gene regulatory systems comprise an important example of a natural system composed of individual elements that interact with each other in a complex fashion, in this case, to regulate and control the production of proteins viable for cell function. Development of analytical and computational tools for the modeling and analysis of gene regulation can substantially help to unravel the mechanisms underlying gene regulation and to understand gene function [55, 56, 57, 58]. This, in turn, can have a profound effect on developing techniques for drug testing and therapeutic intervention for effective treatment of human diseases.

A model of a genetic regulatory network is intended to capture the simultaneous dynamical behavior of various elements, such as transcript or protein levels, for which measurements exist. There have been numerous approaches for modeling the dynamical behavior of genetic regulatory networks, ranging from deterministic to fully stochastic, using either a discrete-time or a continuous-time description of the gene interactions [54]. One way to proceed is to devise theoretical models, for instance, based on systems of differential equations intended to represent as faithfully as possible the joint behavior of all of these constituent elements [59]. The construction of the models, in this case, can be based on existing knowledge of protein-DNA and protein-protein interactions, degradation rates, and other kinetic parameters. Additionally, some measurements focusing on small-scale molecular interactions can be made, with the goal of refining the model. However, global inference of network structure and fine-scale relationships between all the players in a genetic regulatory network is currently an unrealistic undertaking with existing genome-wide measurements produced by microarrays and other high-throughput technologies.

With the understanding that models are intended to predict certain behavior, be it steady-state expression levels of certain groups of genes or functional relationships among a group of genes, we must then develop them with an awareness of the types of available data. For example, it may not be prudent to attempt inferring dozens of continuous-valued rates of change and other parameters in differential equations from only a few discrete-time measurements taken from a population of cells that may not be synchronized with respect to their gene activities (e.g., cell cycle), with a limited knowledge and understanding of the sources of variation due to the measurement technology and the underlying biology. From an engineering perspective, a model should be sufficiently complex to capture the relations necessary for solving the problem at hand, and not so complex that it cannot be reliably estimated from the data. With the advent of microarray technology, a significant effort has been directed at building coarse models of genetic interaction using the limited amount of microarray gene expression data that is usually available. Paradigms that have been considered in this context include Bayesian networks [60], Boolean networks [61], and PBNs (and their extension to PGRNs) [62].

There are two important aspects of every genetic regulatory system that have to be modeled and analyzed. The first is the topology (connectivity structure), and the second is the set of interactions between the elements, the latter determining the dynamical behavior of the system [63, 64, 65]. Exploration of the relationship between topology and dynamics can lead to valuable conclusions about the structure, behavior, and properties of genetic regulatory systems [66, 67].

In a discrete-time functional network, the state of a gene at time $t + 1$ is considered to be a function of a set of genes in a *regulatory set* at time t. The connectivity of the network is defined by the collection of regulatory sets and the interactions are defined by the functions, which are often called *predictors*. A predictor must be designed from data, which *ipso facto* means that it is an approximation of the predictor whose action one would actually like to model. The precision of

the approximation depends on the design procedure and the sample size. Even for a relatively small number of predictor genes, good design can require a very large sample; however, one typically has a small number of microarrays. The problems of classifier design apply essentially unchanged when learning predictors from sample data. To be effectively addressed, they need to be approached within the context of constraining biological knowledge, since prior knowledge significantly reduces the data requirement.

The oldest model for gene regulation is the Boolean network [61, 68, 69, 70, 71]. In a Boolean network, each gene is represented by a binary value, 0 or 1, indicating whether it is down- or up-regulated, and each gene value at the next time point is determined by a function of the gene values in its regulatory set. The action of the network is deterministic and after some finite time, it will settle into an attractor, which is a set of states though which it will endlessly cycle. The Boolean model has recently been extended so that instead of a single predictor function, each gene has a set of predictor functions, one of which is chosen at each time point. This extension results in the class of PBNs [62, 72]. In the early PBN papers, regulatory sets were chosen based on the coefficient of determination, which measures the degree to which the prediction of a target's random variable is improved by observation of the variables in the regulatory set relative to prediction of the target variable using only statistical information concerning the target variable itself [73, 74, 75]. If the predictor choice is random at each time point, then the network is said to be instantaneously random; the predictor is held fixed and only allowed to switch depending on some binary random variable, then the network is said to be context sensitive. The latter case results in a family of Boolean networks composing the PBN, with one of the constituent networks governing gene activity for some period of time. This reflects the effect of latent variables, not incorporated into the model. A PGRN has the same structure as a PBN except that each gene may take on a value within a discrete interval $[0, r]$, with r not being constrained to 0 or 1.

A key objective of network modeling is to use the network to design different approaches for affecting the evolution of the gene state vector over time—for instance, in the case of cancer to drive the network away from states associated with cell proliferation. There have been a number of studies regarding intervention in the context of PBNs . These include resetting the state of the PBN, as necessary, to a more desirable initial state and letting the network evolve from there [76] and manipulating external (control) variables that affect the transition probabilities of the network and can, therefore, be used to desirably affect its dynamic evolution over a finite-time horizon [77, 78]. The latter approach is particularly promising because it involves the use of automatic control theory to derive optimal treatment strategies over time—for instance, using dynamic programming.

Overview of the book

This edited book provides an up-to-date and tutorial-level overview of genomic signal processing (GSP) and statistics. Written by an interdisciplinary team of

authors, the book is accessible to researchers in academia and industry, who are interested in cross-disciplinary areas relating to molecular biology, engineering, statistics, and signal processing. Our goal is to provide audiences with a broad overview of recent advances in the important and rapidly developing GSP discipline.

In the following, we give a brief summary of the contents covered in this book. The book consists of twelve book chapters.

(i) In the first part, we focus on signal processing and statistics techniques in sequence analysis. In "Representation and analysis of DNA sequences," by Paul Dan Cristea, the author presents results in the analysis of genomic information at the scale of whole chromosomes or whole genomes based on the conversion of genomic sequences into genomic signals, concentrating on the phase analysis.

(ii) In the second part, we focus on signal processing and statistics methodologies in gene selection: classification, clustering, and data extraction. In "Gene feature selection," by Ioan Tabus and Jaakko Astola, the authors overview the classes of feature selection methods, and focus specially on microarray problems, where the number of measured genes (factors) is extremely large, in the order of thousands, and the number of relevant factors is much smaller. Classification plays an important role in genomic signal analysis. In "Classification," by Ulisses Braga-Neto and Edward Dougherty, the authors present various techniques in classification, including classifier design, regularization, and error estimation. In "Clustering: revealing intrinsic dependencies in microarray data," by Marcel Brun, Charles D. Johnson, and Kenneth S. Ramos, the authors address clustering algorithms, including interpretation, validation, and clustering microarray data. In "From biochips to laboratory-on-a-chip system," by Lei Wang, Hongying Yin, and Jing Cheng, the authors review various aspects related to biochips with different functionality and chip-based integrated systems.

(iii) In the third part, we focus on signal processing in genomic network modeling and analysis. In "Modeling and simulation of genetic regulatory networks by ordinary differential equations," by Hidde de Jong and Johannes Geiselmann, the authors review various methods for modeling and simulating genetic regulatory network and propose differential equations for regulatory network modeling. In "Modeling genetic regulatory networks with probabilistic Boolean networks," by Ilya Shmulevich and Edward R. Dougherty, the authors present a recently proposed mathematical rule-based model, the probabilistic Boolean networks (PBNs), to facilitate the construction of gene regulatory networks. In "Bayesian networks for genomic analysis," by Paola Sebastiani, Maria M. Abad, and Marco F. Ramoni, the authors show how to apply Bayesian networks in analyzing various types of genomic data, from genomic markers to gene expression data. In "Statistical inference of transcriptional regulatory networks," by Xiaodong Wang, Dimitris Anastassiou, and Dong Guo, the authors present parameter estimation methods for known network structures, including equation-based methods and Bayesian methods. They also discuss Bayesian techniques for inferring network structures.

(iv) In the last part of this book, we focus on microarray imaging, signal processing in systems biology, and applications in disease diagnosis and treatments. In "Compressing genomic and proteomic microarray images for statistical analyses," by Rebecka Jörnsten and Bin Yu, the authors propose a multilayer data structure as the principle for both lossless and lossy compression of microarray images. In "Cancer genomics, proteomics, and clinic applications," by X. Steve Fu, Chien-an A. Hu, Jie Chen, Jane Wang, and K. J. Ray Liu, the authors focus on genomics and proteomics of cancer, and discuss how cutting-edge technologies, like microarray technology and nanotechnology, can be applied in clinical oncology. In "Integrated approach for computational systems biology," by Seungchan Kim, Phillip Stafford, Michael L. Bittner, and Edward B. Suh, the authors address integrated approaches for computational systems biology including biological data and measurement technologies, systems for biological data integration, mathematical and computational tools for computational systems biology, and supercomputing and parallel applications.

Finally, the coeditors would like to thank the authors for their contributions. We hope that readers enjoy this book.

Bibliography

[1] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. E. Darnell, *Molecular Cell Biology*, W. H. Freeman, New York, NY, USA, 4th edition, 2000.

[2] T. A. Brown, *Genomes*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2002.

[3] S. Drăghici, *Data Analysis Tools for DNA Microarrays*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2003.

[4] I. S. Kohane, A. Kho, and A. J. Butte, *Microarrays for an Integrative Genomics*, MIT Press, Cambridge, Mass, USA, 2003.

[5] E. R. Dougherty, I. Shmulevich, and M. L. Bittner, "Genomic signal processing: the salient issues," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 146–153, 2004.

[6] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[7] M. Schena, D. Shalon, R. Heller, A. Chai, P. O. Brown, and R. W. Davis, "Parallel human genome analysis: microarray-based expression monitoring of 1000 genes," *Proc. Natl. Acad. Sci. USA*, vol. 93, no. 20, pp. 10614–10619, 1996.

[8] J. DeRisi, L. Penland, P. O. Brown, et al., "Use of a cDNA microarray to analyse gene expression patterns in human cancer," *Nat. Genet.*, vol. 14, no. 4, pp. 457–460, 1996.

[9] J. L. DeRisi, V. R. Iyer, and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, no. 5338, pp. 680–686, 1997.

[10] D. J. Duggan, M. L. Bittner, Y. Chen, P. S. Meltzer, and J. M. Trent, "Expression profiling using cDNA microarrays," *Nat. Genet.*, vol. 21, Suppl 1, pp. 10–14, 1999.

[11] S. Attoor, E. R. Dougherty, Y. Chen, M. L. Bittner, and J. M. Trent, "Which is better for cDNA-microarray-based classification: ratios or direct intensities," *Bioinformatics*, vol. 20, no. 16, pp. 2513–2520, 2004.

[12] Y. Chen, E. R. Dougherty, and M. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *J. Biomed. Opt.*, vol. 2, no. 4, pp. 364–374, 1997.

[13] M. K. Kerr, M. Martin, and G. A. Churchill, "Analysis of variance for gene expression microarray data," *J. Comput. Biol.*, vol. 7, no. 6, pp. 819–837, 2000.

[14] M. K. Kerr and G. A. Churchill, "Statistical design and the analysis of gene expression microarray data," *Genet. Res.*, vol. 77, no. 2, pp. 123–128, 2001.

[15] M. L. Lee, F. C. Kuo, G. A. Whitmore, and J. Sklar, "Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 18, pp. 9834–9839, 2000.

[16] M. C. Yang, Q. G. Ruan, J. J. Yang, et al., "A statistical method for flagging weak spots improves normalization and ratio estimates in microarrays," *Physiol Genomics*, vol. 7, no. 1, pp. 45–53, 2001.

[17] R. Sasik, E. Calvo, and J. Corbeil, "Statistical analysis of high-density oligonucleotide arrays: a multiplicative noise model," *Bioinformatics*, vol. 18, no. 12, pp. 1633–1640, 2002.

[18] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bittner, P. S. Meltzer, and J. M. Trent, "Ratio statistics of gene expression levels and applications to microarray data analysis," *Bioinformatics*, vol. 18, no. 9, pp. 1207–1215, 2002.

[19] I. Shmulevich and W. Zhang, "Binary analysis and optimization-based normalization of gene expression data," *Bioinformatics*, vol. 18, no. 4, pp. 555–565, 2002.

[20] A. Ben-Dor, N. Friedman, and Z. Yakhini, "Scoring genes for relevance," Tech. Rep. AGL-2000-13, Agilent Laboratories, Palo Alto, Calif, USA, 2000.

[21] L. Wernisch, S. L. Kendall, S. Soneji, et al., "Analysis of whole-genome microarray replicates using mixed models," *Bioinformatics*, vol. 19, no. 1, pp. 53–61, 2003.

[22] Y. Tu, G. Stolovitzky, and U. Klein, "Quantitative noise analysis for gene expression microarray experiments," *Proc. Natl. Acad. Sci. USA.*, vol. 99, no. 22, pp. 14031–14036, 2002.

[23] H. M. Fathallah-Shaykh, M. Rigen, L. J. Zhao, et al., "Mathematical modeling of noise and discovery of genetic expression classes in gliomas," *Oncogene*, vol. 21, no. 47, pp. 7164–7174, 2002.

[24] K. Dobbin, J. H. Shih, and R. Simon, "Statistical design of reverse dye microarrays," *Bioinformatics*, vol. 19, no. 7, pp. 803–810, 2003.

[25] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[26] C. M. Perou, T. Sorlie, M. B. Eisen, et al., "Molecular portraits of human breast tumours," *Nature*, vol. 406, no. 6797, pp. 747–752, 2000.

[27] I. Hedenfalk, D. Duggan, Y. Chen, et al., "Gene-expression profiles in hereditary breast cancer," *N. Engl. J. Med.*, vol. 344, no. 8, pp. 539–548, 2001.

[28] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31 of *Applications of Mathematics (New York)*, Springer-Verlag, New York, NY, USA, 1996.

[29] E. R. Dougherty, "Small sample issues for microarray-based classification," *Comparative and Functional Genomics*, vol. 2, no. 1, pp. 28–34, 2001.

[30] V. N. Vapnik and A. Ya. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory Probab. Appl.*, vol. 16, no. 2, pp. 264–280, 1971.

[31] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[32] T. M. Cover and J. M. van Campenhout, "On the possible orderings in the measurement selection problem," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, no. 9, pp. 657–661, 1977.

[33] S. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: recommendations for practitioners," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 3, pp. 252–264, 1991.

[34] A. K. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 2, pp. 153–158, 1997.

[35] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognition*, vol. 33, no. 1, pp. 25–41, 2000.

[36] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?," *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[37] U. M. Braga-Neto and E. R. Dougherty, "Bolstered error estimation," *Pattern Recognition*, vol. 37, no. 6, pp. 1267–1281, 2004.

[38] C. Sima, U. Braga-Neto, and E. R. Dougherty, "Superior feature-set ranking for small samples using bolstered error estimation," to appear in Bioinformatics.

[39] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, 1998.

[40] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *J. Comput. Biol.*, vol. 6, no. 3-4, pp. 281–297, 1999.
[41] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.
[42] T. Kohonen, *Self-organizing Maps*, vol. 30 of *Springer Series in Information Sciences*, Springer-Verlag, Berlin, Germany, 1995.
[43] A. Flexer, "On the use of self-organizing maps for clustering and visualization," *Intelligent Data Analysis*, vol. 5, pp. 373–384, 2001.
[44] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, no. 3, pp. 803–821, 1993.
[45] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *The Computer Journal*, vol. 41, no. 8, pp. 578–588, 1998.
[46] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 12, pp. 1650–1654, 2002.
[47] G. W. Milligan and M. C. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivariate Behav. Res.*, vol. 21, pp. 441–458, 1986.
[48] R. E. Kass and A. E. Raftery, "Bayes factors," *J. Amer. Statist. Assoc.*, vol. 90, no. 430, pp. 773–795, 1995.
[49] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, no. 10, pp. 977–987, 2001.
[50] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
[51] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2001.
[52] M. K. Kerr and G. A. Churchill, "Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 16, pp. 8961–8965, 2001.
[53] E. R. Dougherty and M. Brun, "A probabilistic theory of clustering," *Pattern Recognition*, vol. 37, no. 5, pp. 917–925, 2004.
[54] H. de Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *J. Comput. Biol.*, vol. 9, no. 1, pp. 67–103, 2002.
[55] D. Endy and R. Brent, "Modelling cellular behaviour," *Nature*, vol. 409, no. 6818, pp. 391–395, 2001.
[56] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: in numero molecular biology," *Nat. Rev. Genet.*, vol. 2, no. 4, pp. 268–279, 2001.
[57] T. Ideker, T. Galitski, and L. Hood, "A new approach to decoding life: systems biology," *Annu. Rev. Genomics Hum. Genet.*, vol. 2, pp. 343–372, 2001.
[58] H. Kitano, "Systems biology: a brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.
[59] T. Mestl, E. Plahte, and S. W. Omholt, "A mathematical framework for describing and analyzing gene regulatory networks," *J. Theor. Biol.*, vol. 176, no. 2, pp. 291–300, 1995.
[60] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 601–620, 2000.
[61] S. A. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, no. 215, pp. 177–178, 1969.
[62] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.
[63] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Modern Phys.*, vol. 74, no. 1, pp. 47–97, 2002.
[64] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.
[65] S. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, 2001.
[66] T. Ideker, V. Thorsson, J. A. Ranish, et al., "Integrated genomic and proteomic analyses of a systematically perturbed metabolic network," *Science*, vol. 292, no. 5518, pp. 929–934, 2001.

[67] D. M. Wolf and F. H. Eeckman, "On the relationship between genomic regulatory element organization and gene regulatory dynamics," *J. Theor. Biol.*, vol. 195, no. 2, pp. 167–186, 1998.
[68] S. A. Kauffman, "The large scale structure and dynamics of gene control circuits: an ensemble approach," *J. Theor. Biol.*, vol. 44, no. 1, pp. 167–190, 1974.
[69] L. Glass and S. A. Kauffman, "The logical analysis of continuous, non-linear biochemical control networks," *J. Theor. Biol.*, vol. 39, no. 1, pp. 103–129, 1973.
[70] S. A. Kauffman, *The Origins of Order: Self-organization and Selection in Evolution*, Oxford University Press, New York, NY, USA, 1993.
[71] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery," *J. Mol. Med.*, vol. 77, no. 6, pp. 469–480, 1999.
[72] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proc. IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.
[73] E. R. Dougherty, M. L. Bittner, Y. Chen, et al., "Nonlinear filters in genomic control," in *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Antalya, Turkey, June 1999.
[74] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Process.*, vol. 80, no. 10, pp. 2219–2235, 2000.
[75] S. Kim, E. R. Dougherty, M. L. Bittner, et al., "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *J. Biomed. Opt.*, vol. 5, no. 4, pp. 411–424, 2000.
[76] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.
[77] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, no. 1-2, pp. 169–191, 2003.
[78] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks: the imperfect information case," *Bioinformatics*, vol. 20, no. 6, pp. 924–930, 2004.

Edward R. Dougherty: Department of Electrical Engineering, Texas A&M University, 3128 TAMU, College Station, TX 77843-3128, USA

Email: edward@ee.tamu.edu

Ilya Shmulevich: The Institute for Systems Biology, 1441 North 34th Street, Seattle, WA 98103-8904, USA

Email: is@ieee.org

Jie Chen: Division of Engineering, Brown University, Providence, RI 02912, USA

Email: jie_chen@brown.edu

Z. Jane Wang: Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

Email: zjanew@eee.ubc.ca

1 Representation and analysis of DNA sequences

Paul Dan Cristea

1.1. Introduction

Data on genome structural and functional features for various organisms is being accumulated and analyzed in laboratories all over the world, from the small university or clinical hospital laboratories to the large laboratories of pharmaceutical companies and specialized institutions, both state owned and private. This data is stored, managed, and analyzed on a large variety of computing systems, from small personal computers using several disk files to supercomputers operating on large commercial databases. The volume of genomic data is expanding at a huge and still growing rate, while its fundamental properties and relationships are not yet fully understood and are subject to continuous revision. A worldwide system to gather genomic information centered in the National Center for Biotechnology Information (NCBI) and in several other large integrative genomic databases has been put in place [1, 2]. The almost complete sequencing of the genomes of several eukaryotes, including man (*Homo sapiens* [2, 3, 4]) and "model organisms" such as mouse (*Mus musculus* [5, 6]), rat (*Rattus norvegicus* [7]), chicken (*Gallus-gallus* [8]), the nematode *Caenorhabditis elegans* [9], and the plant *Arabidopsis thaliana* [10], as well as of a large number of prokaryotes, comprising bacteria, viruses, archeia, and fungi [1, 2, 5, 11, 12, 13, 14, 15, 16, 17, 18, 19], has created the opportunity to make comparative genomic analyses at scales ranging from individual genes or control sequences to whole chromosomes. The public access to most of these data offers to scientists around the world an unprecedented chance to data mine and explore in depth this extraordinary information depository, trying to convert data into knowledge.

The standard symbolical representation of genomic information—by sequences of nucleotide symbols in DNA and RNA molecules or by symbolic sequences of amino acids in the corresponding polypeptide chains (for coding sections)—has definite advantages in what concerns storage, search, and retrieval of genomic information, but limits the methodology of handling and processing genomic information to pattern matching and statistical analysis. This methodological limitation

determines excessive computing costs in the case of studies involving feature extraction at the scale of whole chromosomes, multiresolution analysis, comparative genomic analysis, or quantitative variability analysis [20, 21, 22].

Converting the DNA sequences into digital signals [23, 24] opens the possibility to apply signal processing methods for the analysis of genomic data [23, 24, 25, 26, 27, 28, 29, 30, 31, 32] and reveals features of chromosomes that would be difficult to grasp by using standard statistical and pattern matching methods for the analysis of symbolic genomic sequences. The genomic signal approach has already proven its potential in revealing large scale features of DNA sequences maintained over distances of 10^6–10^8 base pairs, including both coding and noncoding regions, at the scale of whole genomes or chromosomes (see [28, 31, 32], and Section 1.4 of this chapter). We enumerate here some of the main results that will be presented in this chapter and briefly outline the perspectives they open.

1.1.1. Unwrapped phase linearity

One of the most conspicuous results is that the average unwrapped phase of DNA complex genomic signals varies almost linearly along all investigated chromosomes, for both prokaryotes and eukaryotes [23]. The magnitude and sign of the slope are specific for various taxa and chromosomes. Such a behavior proves a rule similar to Chargaff's rule for the distribution of nucleotides [33]—a statistics of the first order, but reveals a statistical regularity of the succession of the nucleotides—a statistics of the second order. As can be seen from equation (1.11) in Section 1.4, this rule states that the difference between the frequencies of positive and negative nucleotide-to-nucleotide transitions along a strand of a chromosome tends to be small, constant, and taxon & chromosome specific. As an immediate practical use of the unwrapped phase quasilinearity rule, the compliance of a certain contig with the large scale regularities of the chromosome to which it belongs can be used for spotting errors and exceptions.

1.1.2. Cumulated phase piecewise linearity in prokaryotes

Another significant result is that the cumulated phase has an approximately piecewise linear variation in prokaryotes, while drifting around zero in eukaryotes. The breaking points of the cumulated phase in prokaryotes can be put in correspondence with the limits of the chromosome "replichores": the minima with the origins of the replication process, and the maxima with its termini.

The existence of large scale regularities, up to the scale of entire chromosomes, supports the view that extragene DNA sequences, which do not encode proteins, still play significant functional roles. Moreover, the fact that these regularities apply to both coding and noncoding regions of DNA molecules indicates that these functionalities are also at the scale of the entire chromosomes. The unwrapped and cumulated phases can be linked to molecule potentials produced by unbalanced hydrogen bonds and can be used to describe "lateral" interaction of a given DNA segment with proteins and with other DNA segments in processes like replication,

transcription, or crossover. An example of such a process is the movement of DNA polymerase along a DNA strand during the replication process, by operating like a "Brownian machine" that converts random molecule movements into an ordered gradual advance.

1.1.3. Linearity of the cumulated phase for the reoriented exons in prokaryotes

A yet other important result is the finding that the cumulated phase becomes linear for the genomic signals corresponding to the sequences obtained by concatenating the coding regions of prokaryote chromosomes, after reorienting them in the same positive direction. This is a property of both circular and linear prokaryote chromosomes, but is not shared by most plasmids. This "hidden linearity" of the cumulated phase suggests the hypothesis of an ancestral chromosome structure, which has evolved into the current diversity of structures, under the pressure of processes linked to species separation and protection.

The rest of this chapter presents the vector and complex representations of nucleotides, codons, and amino acids that lead to the conversion of symbolic genomic sequences into digital genomic signals and presents some of the results obtained by using this approach in the analysis of large scale properties of nucleotide sequences, up to the scale of whole chromosomes.

Section 1.2 briefly describes aspects of the DNA molecule structure, relevant for the mathematical representation of nucleotides. Section 1.3 presents the vector (3D, tetrahedral) and the complex (2D, quadrantal) representations of nucleotides (Section 1.3.1), codons, and amino acids (Section 1.3.2). It is shown that both the tetrahedral and the quadrantal representations are one-to-one mappings, which contain the same information as the symbolic genomic sequences. Their main advantage is to reveal hidden properties of the genetic code and to conveniently represent significant features of genomic sequences.

Section 1.4 presents the phase analysis of genomic signals for nucleotide sequences and gives a summary of the results obtained by using this methodology. The study of complex genomic signals using signal processing methods facilitates revealing large scale features of chromosomes that would be otherwise difficult to find.

Based on the phase analysis of complex genomic signals, Section 1.5 presents a model of the "patchy" longitudinal structure of chromosomes and advances the hypothesis that it derives from a putative ancestral highly ordered chromosome structure, as a result of processes linked to species separation and specificity protection at molecular level. As mentioned above, it is suggested that this structure is related to important functions at the scale of chromosomes.

In the context of operating with a large volume of data at various resolutions and visualizing the results to make them available to humans, the problem of data representability becomes critical. Section 1.6 presents a new approach to this problem using the concept of data scattering ratio on a pixel. Representability analysis

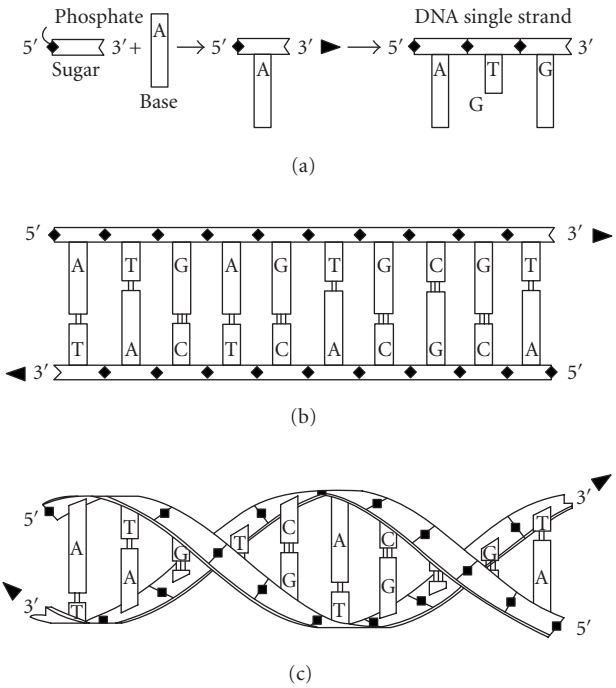


Figure 1.1. Schematic model of the DNA molecule.

is applied to several cases of standard signals and genomic signals. It is shown that the variation of genomic data along nucleotide sequences, specifically the cumulated and unwrapped phase, can be visualized adequately as simple graphic lines for low and large scales, while for medium scales (thousands to tens of thousands of base pairs), the statistical descriptions have to be used.

1.2. DNA double helix

This section gives a brief summary of the structure, properties, and functions of DNA molecules, relevant to building mathematical representations of nucleotides, codons, and amino acids and in understanding the conditions to be satisfied by the mappings of symbolic sequences to digital signals. The presentation is addressed primarily to readers with an engineering background, while readers with a medical or biological training can skip this section.

The main nucleic genetic material of cells is represented by DNA molecules that have a basically simple and well-studied structure [34]. The DNA double helix molecules comprise two antiparallel intertwined complementary strands, each a helicoidally coiled heteropolymer. The repetitive units are the nucleotides, each consisting of three components linked by strong covalent bounds: a monophosphate group linked to a sugar that has lost a specific oxygen atom—the deoxyribose, linked in turn to a nitrogenous base, as schematically shown in Figure 1.1

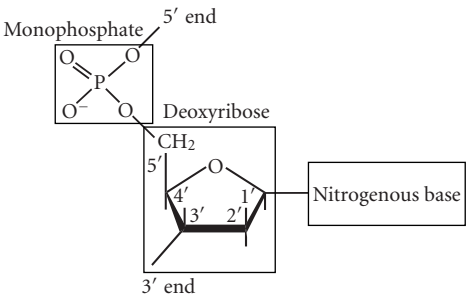


Figure 1.2. Structure of a nucleotide.

and detailed in Figure 1.2. Only four kinds of nitrogenous bases are found in DNA: thymine (T) and cytosine (C)—which are pyrimidines, adenine (A) and guanine (G)—which are purines. As can be seen in Figures 1.3 and 1.5, purine molecules consist of two nitrogen-containing fused rings (one with six atoms and the other with five), while pyridmidine molecules have only a six-membered nitrogen-containing ring. In a ribonucleic acid (RNA) molecule, apart of the replacement of deoxyribose with ribose in the helix backbone, thymine is replaced by uracil (U), a related pyrimidine. As shown in Figure 1.3, along the two strands of the DNA double helix, a pyrimidine in one chain always faces a purine in the other, and only the base pairs T−A and C−G exist. As a consequence, the two strands of a DNA helix are complementary, store the same information, and contain exactly the same number of A and T bases and the same number of C and G bases. This is the famous first Chargaff's rule [33], found by a chemical analysis of DNA molecules, before the crucial discovery of the double helix structure of DNA by Watson and Crick [34], and fully confirmed by this model. The simplified model in Figure 1.1 shows schematically how the nucleotides are structured, the single and double stranded DNA molecules, and gives a sketchy representation of the DNA secondary structure—the double helix resulting from the energy minimization condition. The figure does not show other significant features of the DNA longitudinal structure, such as the major and minor grooves. The hydrogen bonds within the complementary base pairs keep the strands together. When heating double stranded DNA at temperatures around 90°C, the hydrogen bonds melt and the two strands separate, resulting in "DNA denaturation." If lowering again the temperature, the reverse process—called "DNA renaturation"—reestablishes the double helix structure. The pair A−T contains only two hydrogen bonds, while the couple C−G contains three hydrogen bonds, so that the second link is stronger. This is reflected in an increased melting temperature for DNA molecules with a higher C-G concentration. Along each chain, there is a well-defined positive direction, given by the 5′ to 3′ direction in which the strand can grow by the addition of new nucleotides. The growth of a DNA chain is quite a complex process requiring the fulfillment of several conditions, from which we mention only the most important few. The normal process of growing a new DNA single-chain molecule is

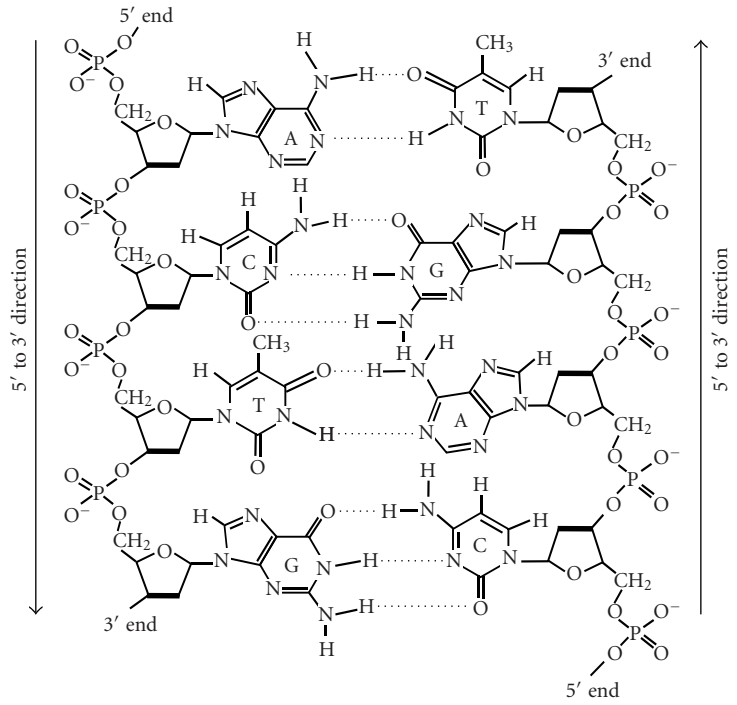


Figure 1.3. The chemical model of a double-stranded DNA molecule.

the replication, in which an existing (complementary) strand is used as a template, along which moves the DNA polymerase—the enzyme that performs the replication, adding to the growing chain nucleotides complementary to the ones in the template. A primer is also required; that is, the DNA polymerase can only prolong an already existing strand, by interacting with both the template strand and the strand to which it adds the nucleotide. The replication process consumes energy, so that the molecules that are needed by DNA polymerase to perform the addition of a nucleotide to the chain are not directly the nucleosine monophosphates, the monomers in the DNA strand, but the nucleosine triphosphates, which contain three phosphate groups and have the necessary free energy stored in the two phosphoanhydride bonds. Figure 1.4 gives the chemical model of adenosine triphosphate (ATP), the nucleosine triphosphate needed to add an adenine nucleotide to a DNA strand. The energy is released by the hydrolysis of the phosphoanhydride bonds and the loss of the two additional phosphate groups. This mechanism is so successful that nature uses ATP molecules not only for the replication of DNA but also for any other biochemical process that requires additional energy, ATP being the "molecular currency" of intracellular energy transfer. In the synthesis of nucleic acids, the ATP to AMP conversion mechanism imposes the 5′ to 3′ positive direction for the growth of DNA strands.

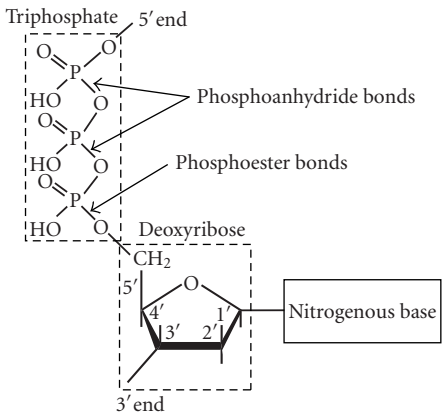


Figure 1.4. The chemical structure of ATP, precursor of the adenoside (adenosine monophosphate)—one of the nucleotides, and the most ubiquituos source of biochemical energy at molecular level.

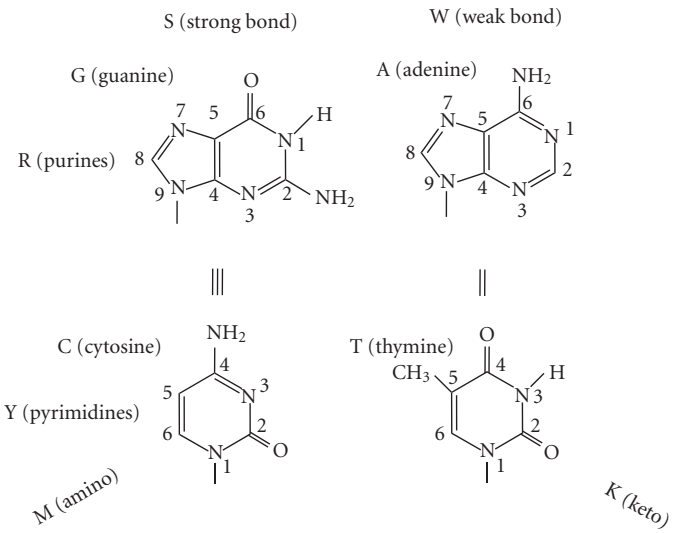


Figure 1.5. Class structure of nitrogenous bases.

The entities in the nucleotide chains that encode polypeptides, that is, specify the primary structure of proteins, are called genes. Genes are divided into exons—coding regions, interrupted by introns—noncoding regions. According to the current GenBank statistics [2], exons in the human genome account for about 3% of the total DNA sequence, introns for about 30%, and intergenic regions for the remaining 67%. Different methodologies produce different results in what concerns the number and size of the coding and noncoding regions. Based on mRNA and EST (Expressed Sequence Tags) studies, human genes contain on the

Table 1.1. The genetic code.

First position in codon		Second position in codon: T			C			A			G			Third position in codon
	T	TTT	Phe	[F]	TCT	Ser	[S]	TAT	Tyr	[Y]	TGT	Cys	[C]	T
		TTC	Phe	[F]	TCC	Ser	[S]	TAC	Tyr	[Y]	TGC	Cys	[C]	C
		TTA	Leu	[L]	TCA	Ser	[S]	TAA	*Ter*	[*end*]	TGA	*Ter*	[*end*]	A
		TTG	Leu	[L]	TCG	Ser	[S]	TAG	*Ter*	[*end*]	TGG	Trp	[W]	G
	C	CTT	Leu	[L]	CCT	Pro	[P]	CAT	His	[H]	CGT	Arg	[R]	T
		CTC	Leu	[L]	CCC	Pro	[P]	CAC	His	[H]	CGC	Arg	[R]	C
		CTA	Leu	[L]	CCA	Pro	[P]	CAA	Gln	[Q]	CGA	Arg	[R]	A
		CTG	Leu	[L]	CCG	Pro	[P]	CAG	Gln	[Q]	CGG	Arg	[R]	G
	A	ATT	Ile	[I]	ACT	Thr	[T]	AAT	Asn	[N]	AGT	Ser	[S]	T
		ATC	Ile	[I]	ACC	Thr	[T]	AAC	Asn	[N]	AGC	Ser	[S]	C
		ATA	Ile	[I]	ACA	Thr	[T]	AAA	Lys	[K]	AGA	Arg	[R]	A
		ATG	Met	[M]	ACG	Thr	[T]	AAG	Lys	[K]	AGG	Arg	[R]	G
	G	GTT	Val	[V]	GCT	Ala	[A]	GAT	Asp	[D]	GGT	Gly	[G]	T
		GTC	Val	[V]	GCC	Ala	[A]	GAC	Asp	[D]	GGC	Gly	[G]	C
		GTA	Val	[V]	GCA	Ala	[A]	GAA	Glu	[E]	GGA	Gly	[G]	A
		GTG	Val	[V]	GCG	Ala	[A]	GAG	Glu	[E]	GGG	Gly	[G]	G

average 3 and 10 exons, respectively, having an average length of 631 bp/262 bp and being separated by introns with average length 6,106 bp/5,420 bp. But there is a very large dispersion, with exon length ranging from just 1 bp/6 bp, up to 12,205 bp/17,105 bp. Minimum intron length is 17 bp/1 bp, while the maximum value reaches 482,576 bp/1,986,943 bp. Protein coding regions are rich in C and G, while intergene (noncoding) regions are rich in T and A.

Protein coding is governed by the genetic code that gives the mapping of codons—triplets of successive nucleotides in the corresponding reading frame in the exons—to the 20 amino acids found in the polypeptide chains and to the terminator that marks the end of an encoding segment. The genetic code is universal, applying to all known nuclear genetic material, DNA, mRNA, and tRNA, and encompasses animals (including humans), plants, fungi, bacteria, and viruses, with only small variations in mitochondria, certain eubacteria, cilliate, fungi, and algae [2]. From Table 1.1, which gives the standard genetic code, it can be seen that there is a large redundancy (degeneration) of the genetic code, as there are $4^3 = 64$ codons to specify only 21 distinct outputs. The redundancy is distributed unevenly among the outputs: there are amino acids encoded by one (2 instances), two (9 instances), three (one instance), four (5 instances), or six (3 instances) distinct codons, while the terminator is encoded by three codons. Most genes start with the codon ATG that also encodes the amino acid methionine.

The codon—amino acid mapping comprises two steps: (1) the *transcription*, in which a specific enzyme, called transcriptase, copies a section of the DNA template into a complementary mRNA (messenger RNA) molecule, in the presence of a mixture of the four ribonucleotides (ATP, UTP, GTP, and CTP), and (2) the *translation*, in which the actual mapping of the codons in the mRNA to amino

acids is performed by ribosomes, after *slicing*—the editing of mRNA by the excision of all introns and the joining of all exons. Quite surprisingly, the number of nucleotides in an exon is not necessarily a multiple of three, that is, an exon does not necessarily comprise an integer number of codons. The amino acids for the protein are brought to the site by tRNA (transfer RNA) molecules, each having a nucleotide triplet which binds to the complementary sequence on the mRNA. Each of the 20 amino acids is brought by a specific tRNA. In fact, there are at least 23 tRNAs for the 20 amino acids, as it will be discussed in the following in relation with the representation of the genetic code. There is a sharp contrast between the deceivingly simple structure of DNA nucleotide chains—unbranched linear code written in a four-letter alphabet, and the overwhelming complexity of the protein 3D structure built of twenty amino acids. As mentioned, there are only about 30 000 genes in the human genome, but millions of proteins, many of them transitory. Nevertheless, the nucleotide chains and the proteins are the bearers of essentially the same genetic information.

1.3. Conversion of genomic sequences into genomic signals

The conversion of genomic sequences from the symbolic form given in the public genomic databases [1, 2] into digital genomic signals allows using signal processing procedures for processing and analyzing genomic data. We have investigated a large number of mappings of symbolic genomic data to digital genomic signals and we have compared how the structure of the genomic code was highlighted by the various representations and how the features of DNA sequences were revealed by the resulting digital signals [25, 26, 27, 28, 29, 30, 31, 32]. Such a representation has to be both truthful and unbiased. The mapping is truthful if all biologically relevant characteristics of the represented objects are expressed in corresponding mathematical properties of the samples in the resulting digital signal. The mapping is unbiased if none of the features belonging to the mapping itself, but without correspondent in the properties of the initial sequence, is introduced as artifacts. The representation must also be simple enough to allow fast and computationally effective conversion and to provide an output readable for a human operator. The last request favors representations with low dimensions of the output, preferably 1D or 2D. This section briefly presents the digital representation of nucleotides starting from the essential features of DNA sequences. A detailed study of the symbolic-to-digital conversion of genomic sequences can be found in [23].

1.3.1. Nucleotide representation

As schematically shown in Figure 1.5, there are three main dichotomies of the nitrogenous bases biochemical properties that allow arranging them in classes: (1) *molecular structure*—A and G are purines (R), while C and T are pyrimidines (Y); (2) *strength of links*—bases A and T are linked by two hydrogen bonds (W—weak bond), while C and G are liked by three hydrogen bonds (S—strong bond);

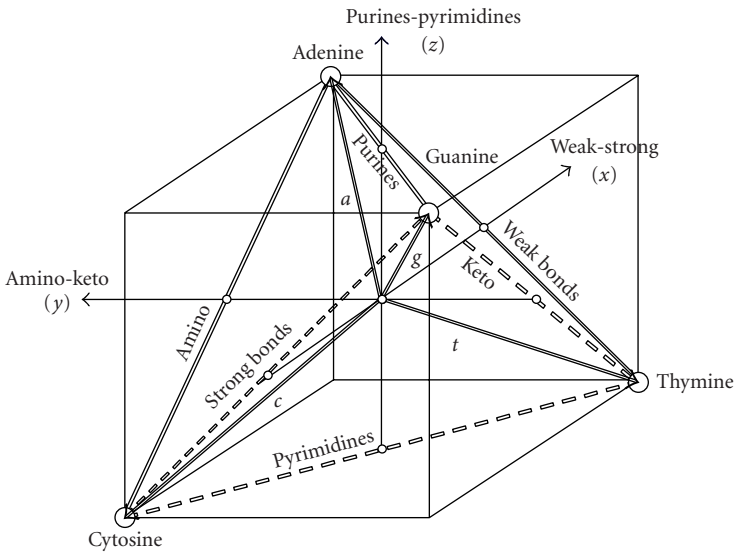


Figure 1.6. Nucleotide tetrahedron.

(3) *radical content*—A and C contain the amino (NH_3) group in the large groove (M class), while T and G contain the keto (C=O) group (K class).

To express the classification of the system of nucleotides in couples of pairs shown in Figure 1.5, we have proposed the nucleotide tetrahedral representation [24] shown in Figure 1.6. The nucleotides are mapped to four vectors symmetrically placed in the 3D space, that is, oriented towards the vertices of a regular tetrahedron. Each of the six edges corresponds to one of the classes comprising a pair of nucleotides. The representation is three dimensional and the axes express the differences "weak minus strong bonds," "amino minus keto," and "purines minus pyrimidines":

$$x = \mathrm{W} - \mathrm{S}, \qquad y = \mathrm{M} - \mathrm{K}, \qquad z = \mathrm{R} - \mathrm{Y}. \tag{1.1}$$

By choosing $\{\pm 1\}$ coordinates for the vertices of the embedding cube, the vectors that represent the four nucleotides take the simple form:

$$\begin{aligned} \vec{a} &= \vec{i} + \vec{j} + \vec{k}, \\ \vec{c} &= -\vec{i} + \vec{j} - \vec{k}, \\ \vec{g} &= -\vec{i} - \vec{j} + \vec{k}, \\ \vec{t} &= \vec{i} - \vec{j} - \vec{k}. \end{aligned} \tag{1.2}$$

This representation is fully adequate for well-defined sequences, when each entry is uniquely specified. Such sequences are given in the large integrative genomic databases, which provide a single curated standard sequence with respect to which single nucleotide polymorphisms (SNPs) or other variations are defined. But, when working with experimental data that can have ambiguous or multiple values for some entries in the sequence, caused by either noise, or by the true variability within the population for which the genome is sequenced, the IUPAC conventions [2] have to be used. Apart of the symbols for the nucleotides (A, C, G, T), IUPAC conventions include symbols for the classes mentioned at the beginning of this section (S, W, R, Y, M, K), as well as for classes comprising three nucleotides (B = {C, G, T} =∼ A, D = {A, G, T} =∼ C, H = {A, C, T} =∼ G, V = {A, C, G =∼ T}), or all four nucleotides (i.e., unspecified nucleotide, N). The corresponding vector representation is shown in Figure 1.7, in which the additional vectors are given by:

$$\begin{aligned}
\vec{w} &= \frac{\vec{a}+\vec{t}}{2} = \vec{i}, \\
\vec{s} &= \frac{\vec{c}+\vec{g}}{2} = -\vec{i}, \\
\vec{m} &= \frac{\vec{a}+\vec{c}}{2} = \vec{j}, \\
\vec{k} &= \frac{\vec{g}+\vec{t}}{2} = -\vec{j}, \\
\vec{r} &= \frac{\vec{a}+\vec{g}}{2} = \vec{k}, \\
\vec{y} &= \frac{\vec{c}+\vec{t}}{2} = -\vec{k}, \\
\vec{b} &= \frac{\vec{c}+\vec{g}+\vec{t}}{3} = -\frac{\vec{a}}{3}, \\
\vec{d} &= \frac{\vec{g}+\vec{t}+\vec{a}}{3} = -\frac{\vec{c}}{3}, \\
\vec{h} &= \frac{\vec{t}+\vec{a}+\vec{c}}{3} = -\frac{\vec{g}}{3}, \\
\vec{u} &= \frac{\vec{a}+\vec{c}+\vec{g}}{3} = -\frac{\vec{t}}{3}.
\end{aligned} \tag{1.3}$$

The dimensionality of the representation can be reduced to two, by projecting the nucleotide tetrahedron on an adequately chosen plane. This plane can be put in correspondence with the complex plane, so that a complex representation of the nucleotides is obtained. The choice of the projection plane is determined by the features that have to be conserved as being most relevant in the given context. For the study of large scale features of DNA sequences and for other similar problems, we have found that the separation within the amino-keto classes is less significant as compared to the strong-weak and purine-pyrimidine dichotomies.

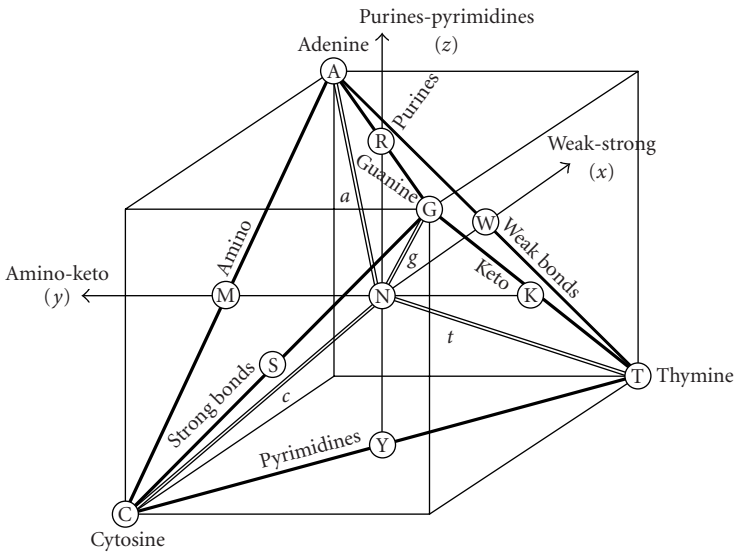


Figure 1.7. IUPAC symbols 3D representation.

This corresponds to the projection on the plane xOz and expresses the S−W and Y−R dichotomies. The resulting complex quadrantal representation of the nucleotides is given in Figure 1.8, in which the pairs of nucleotides are grouped in the six above-mentioned classes, while the corresponding analytical expressions are given in the equations:

$$\begin{aligned} a &= 1 + j, \\ c &= -1 - j, \\ g &= -1 + j, \\ t &= 1 - j. \end{aligned} \tag{1.4}$$

In this representation the complementarity of the pairs of bases A−T and C−G, respectively, is expressed by the symmetry with respect to the real axis (the representations are complex conjugates: $t = a^*, g = c^*$), while the purine/pyrimidine pairs have the same imaginary parts. We have investigated several other representations, but the complex representation given by (1.4) has shown most advantages.

It should be noted that both the vector (3D, tetrahedral) and the quadrantal (2D, complex) nucleotide representations shown above, as well as the real (1D) representation to be discussed in the following, are one-to-one mappings that allow rebuilding the initial symbolic sequence from the vector, complex or real genomic signals. The information is wholly conserved and so are all the features and properties of the initial sequences. Nevertheless, there are significant differences in what concerns the expression of the various significant features and how accessible

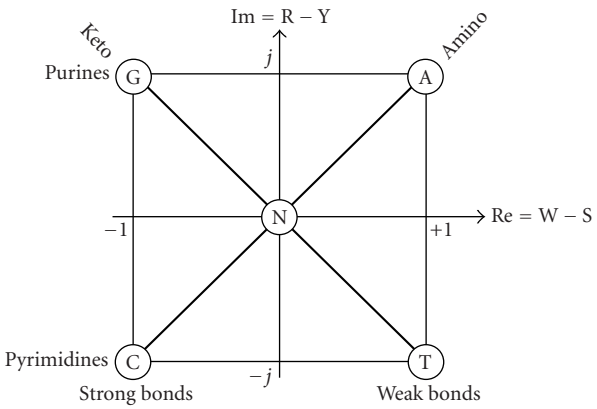


Figure 1.8. Nucleotide complex representation.

or directly readable for a human agent these features become. As in many other cases, a good representation of a system is an important part in solving various problems related to that system.

The projection of the vectors in Figure 1.7 on the same xOz plane provides the complex representation of the IUPAC symbols given in Figure 1.9 and expressed by the equations:

$$\begin{aligned} w &= 1, \\ y &= -j, \\ s &= -1, \\ r &= j, \\ k &= m = n = 0, \\ d &= \frac{1}{3}(1+j), \\ h &= \frac{1}{3}(1-j), \\ b &= \frac{1}{3}(-1-j), \\ v &= \frac{1}{3}(-1+j). \end{aligned} \tag{1.5}$$

As mentioned above, it is possible to further reduce the dimensionality of the representation of nucleotide, codon, and amino acid sequences by using a real one-dimensional mapping. The digits $\{0, 1, 2, 3\}$ can be attached to the four nucleotides. The three-base codons are interpreted as three-digit numbers written in base four, that is, the codons along the DNA strands are mapped to the numbers $\{0, 1, 2, \ldots, 63\}$. Actually, a whole DNA sequence can be seen as a very large number written in base four. Nevertheless, it corresponds better to the biological reality to interpret each codon as a distinct sample of a digital genomic signal distributed

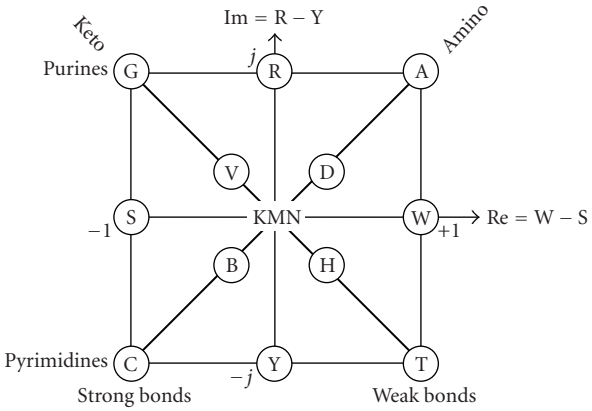


Figure 1.9. IUPAC symbols complex representation.

Table 1.2. Real representation of nucleotides to digits in base four.

Pyrimidines	Purines
Thymine = T = 0	Adenine = A = 2
Cytosine = C = 1	Guanine = G = 3

along the DNA strand. There are 4! = 24 choices for attaching the digits 0–3 to the bases A, C, G, T. The optimal choice given in Table 1.2 results from the condition to obtain the most monotonic mapping of the codons 0–63 to the amino acids plus the terminator 0–20, leading to best autocorrelated intergene genomic signals [23].

1.3.2. Codon and amino acid representation

The tetrahedral (3D), complex (2D), and real (1D) representations of nucleotides can be naturally extended for the representation of codons and amino acids.

A codon consists of a sequence of three nucleotides:

$$X = B_2B_1B_0, \quad B_i \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\};\ i = 0, 1, 2, \tag{1.6}$$

situated in a coding area of a DNA molecule, that is, in an exon, and having the start aligned to the correct open reading frame (ORF). There are six possible ORFs, three in each direction of the DNA strand, shifted with a nucleotide from each other.

The codon can be seen as a word of three letters, taken from a four-letter alphabet. The codon can also be seen as a number written in a certain base, using the four digits B_i. For the vectorial (tetrahedral) representation of nucleotides, we have chosen the base two and the four-vector digits having the values given in

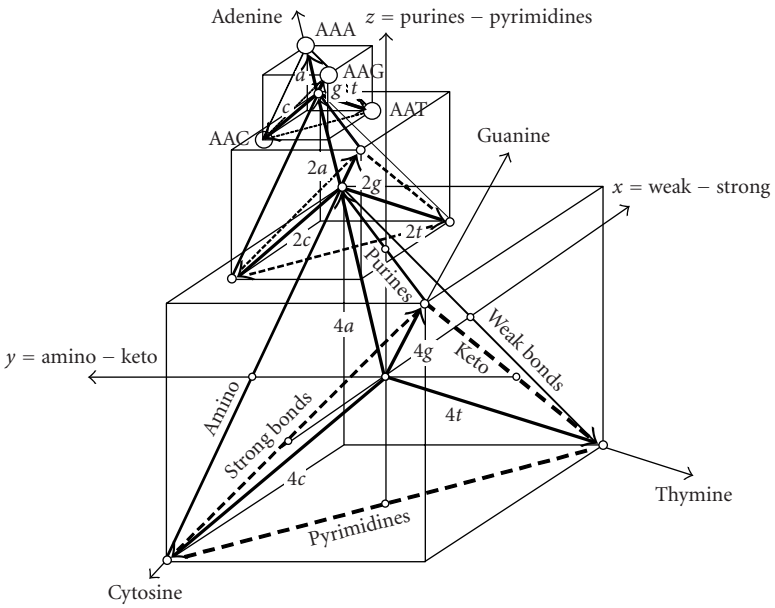


Figure 1.10. Example of the vector representation of codons.

equation (1.2). Correspondingly, the codon X is mapped to the vector:

$$\vec{x} = 2^2\vec{b}_2 + 2^1\vec{b}_1 + 2^0\vec{b}_0, \quad \vec{b}_i \in \{\vec{a}, \vec{c}, \vec{g}, \vec{t}\};\ i = 0, 1, 2. \tag{1.7}$$

This is a natural extension of the concept of *numeration system* to vectorial (and complex) numbers. The vectorial conversion procedure is repeated for each of the three nucleotides in a codon, treating them as digits of a three-digit number written in base two: the vector corresponding to the third, that is, the last nucleotide in the codon (the least significant digit) is multiplied by 1, the vector corresponding to the second base in the codon by 2, and the vector corresponding to the first base of the codon (the most significant digit) by $2^2 = 4$. This results in the codon vectorial representation illustrated in Figure 1.10 for the special cases of the codons AAA $(4\vec{a}+2\vec{a}+\vec{a})$ and AAG $(4\vec{a}+2\vec{a}+\vec{g})$—encoding lysine, and AAC $(4\vec{a}+2\vec{a}+\vec{c})$ and AAT $(4\vec{a}+2\vec{a}+\vec{t})$—encoding asparagine. Applying the same rule for all the 64 codons, the codon tetrahedral representation in Figure 1.11 is obtained [24]. The first nucleotide in a codon selects one of the four *first-order* 16-*codon tetrahedrons* that form together the *zero-order tetrahedron* of the overall genetic code, the second nucleotide selects one of the *second-order* 4-*codon tetrahedrons* that compose the already selected first-order tetrahedron and, finally, the third nucleotide identifies one of the vertices. In this way, each of the codons is attached to one of the vertices in a resulting three-level fractal-like tetrahedron structure. Taking into account the codon-to-amino acid mapping imposed by the genetic code, the amino acids encoded by the codons can be assigned to one or

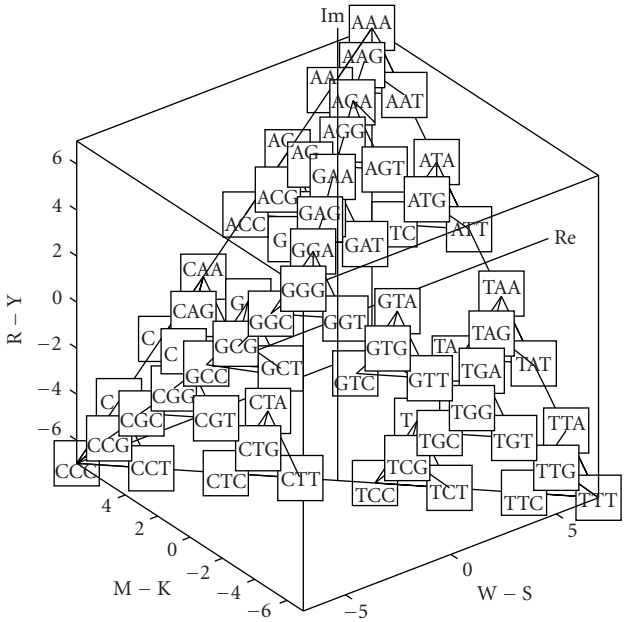


Figure 1.11. Codon tetrahedral representation.

several of the 64 vertices, as shown in Figure 1.12. It turns out that the tetrahedron representation of the genomic code, as well as the mathematical descriptions based on it, reflects better the metric structure of the genomic code. Specifically, the codons that correspond to the same amino acid are mapped in neighboring points, so that related codons are clustered. Moreover, the degeneration is basically restricted to the second-order tetrahedrons and most pairs of interchangeable nucleotides are distributed on the edges parallel to the pyrimidines and purines directions. The tetrahedron representation has also the advantage to naturally determine putative ancestral coding sequences by the simple passage to a lower-level tetrahedron. Thus, the tetrahedron representation grasps some essential features of the genetic code which appear as symmetries and regularities of the resulting 3D image. To make the nucleotide and codon sequences easy to read for a human observer, the three axes of the representation space can be assigned to the three basic color components of the RGB—red, green, blue system [35]. Consequently, each point in the representation space—each nucleotide in the case of Figure 1.6, or each IUPAC symbol in the case of Figure 1.7, corresponds to a distinct hue. This approach is useful for the fast visual exploration of DNA sequences at the nucleotide level and can be extended at the codon (Figure 1.11) and amino acid levels (Figure 1.12). The superposition of the codon tetrahedron and of the amino acid tetrahedron, as shown in Figure 1.13, is the 3D equivalent of a periodic table for the genomic code. This representation gives a better image of the regularities of the genomic code and allows sensing of some aspects of its early evolution before

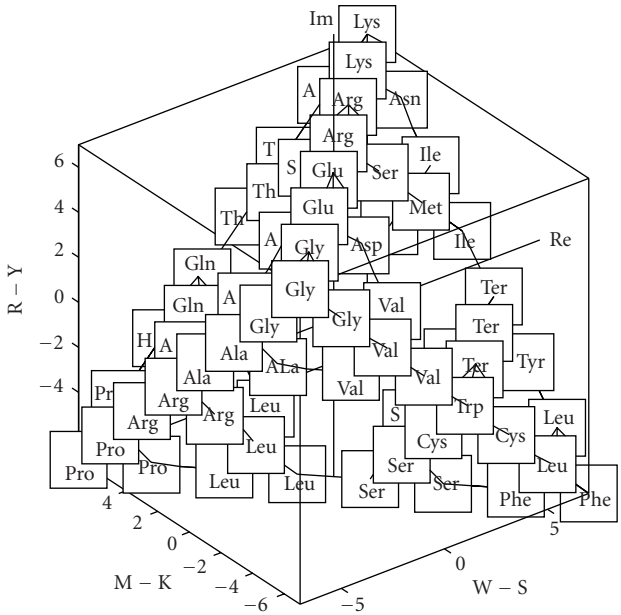


Figure 1.12. Amino acid tetrahedral representation.

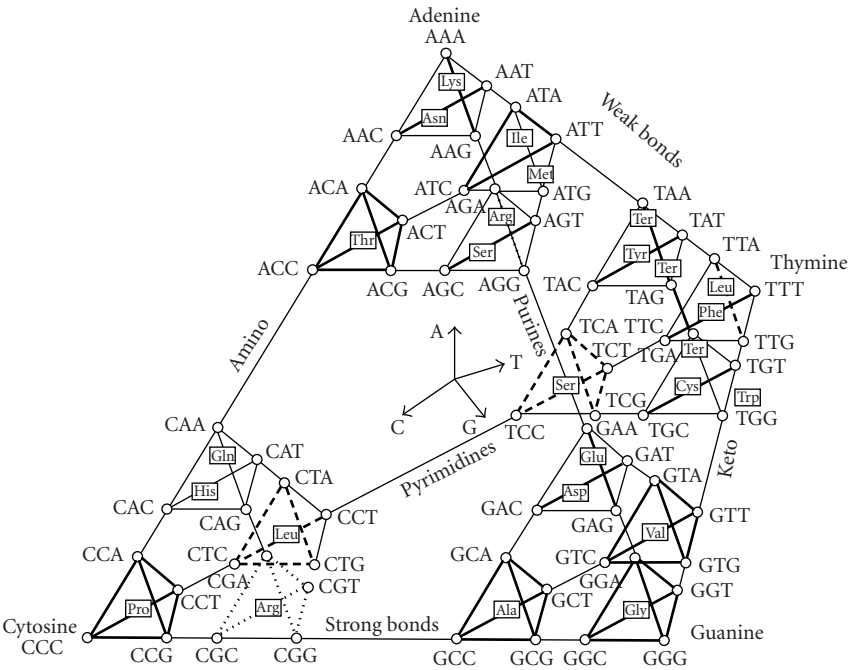
reaching the current frozen state. It is especially remarkable that the different representations of an amino acid, resulting from the redundancy of the genetic code, are mapped in neighboring points of the codon tetrahedron, with the exception of the three instances of amino acids degenerated of order six, for which none of the investigated mapping can obtain the contiguity of the representations. It must be mentioned, though, that for each of these amino acids, there are two distinct tRNA (giving the total of 23 tRNAs for the 20 amino acids), each encoded by the codons of the same cluster.

A complex representation of the codons can be obtained in a similar way, by considering the codon as a three-digit number in base two and by using the complex representations (1.4) of its nucleotides as the digits:

$$x = 2^2 b_2 + 2^1 b_1 + 2^0 b_0, \quad b_i \in \{a, c, g, t\};\ i = 0, 1, 2. \tag{1.8}$$

Again, relation (1.8) results from (1.7) by the projection on the xOz plane, and by taking Ox as the real axis and Oz as the imaginary one. Relation (1.8) can also be seen as representing the nucleotides in two (bipolar) binary systems, with mutually orthogonal complex units.

For methionine, to which corresponds the codon ATG that starts any gene, the vector representation is $\vec{M} = 4\vec{a} + 2\vec{t} + \vec{g} = 5\vec{i} + \vec{j} + 3\vec{k}$, while the complex representation is $M = 4a + 2t + g = 5 + 3j$. For the example in Figure 1.10, the corresponding complex representation is given in Figure 1.14.



Ala - alanine	Gly - glycine	Pro - proline
Arg - arginine	Ile - isoleucine	Ser - serine
Asn - asparagine	His - histidine	Thr - thereonine
Asp - aspartic acid	Leu - leucine	Trp - tryptophan
Cys - cysteine	Lys - lysine	Tyr - tyrosine
Gln - glutamine	Met - methionine	Val - valine
Glu - glutamic acid	Phe - phenylalanine	Ter - terminator

Figure 1.13. Genetic code vectorial representation.

Applying the same method for the 64 codons, the complete mapping of all the codons to the complex plane given in Figure 1.15 is obtained [23]. This is the projection on the xOz plane of the codon tetrahedron in Figure 1.11. Figure 1.16 shows the mapping of the amino acids to the complex plane and can be obtained either by applying the genetic code on the complex representation of the codons in Figure 1.11, or by projecting the amino acid tetrahedron in Figure 1.12 on the xOz plane. The superposition of the codon and amino acid complex representations in Figure 1.17 gives a complex (2D) representation of the genomic code. The clustering of the codons corresponding to the same tRNAs is obvious, and this corresponds in 17 of the cases with the clustering of all the codons corresponding to the same amino acid in a single contiguous domain. Only for arginine, leucine, and serine, each of them encoded by six codons, the corresponding domains are splitted in two noncontiguous subdomains comprising four and, respectively, two codons. It is interesting to mention that the clustering refers not

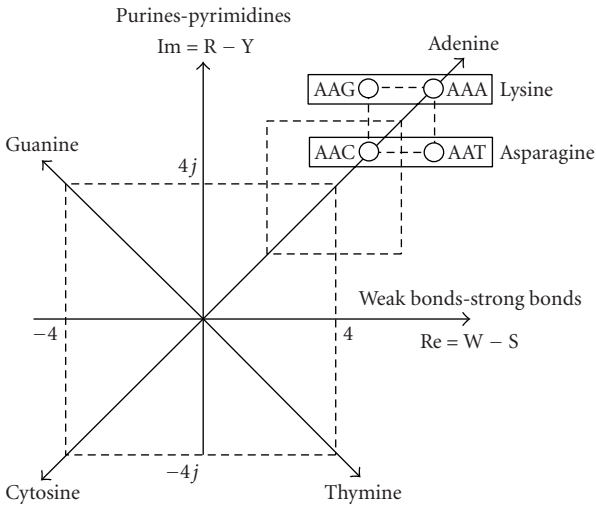


Figure 1.14. Codon complex representation.

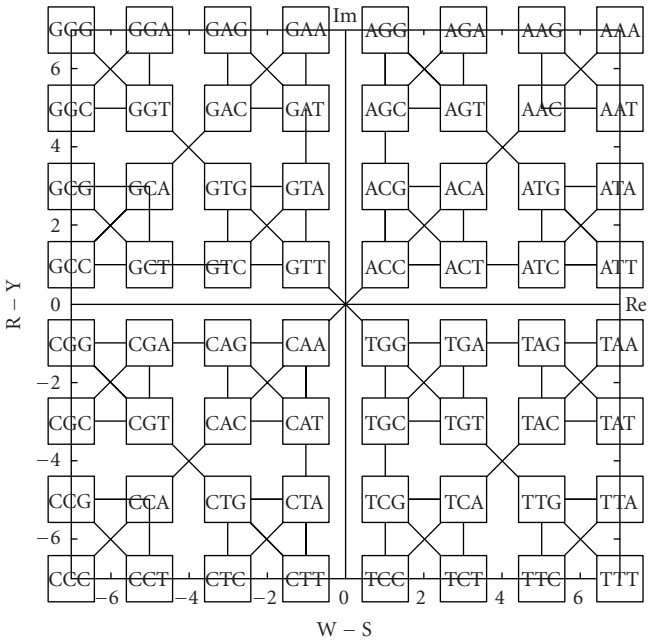


Figure 1.15. Mapping of the codons on the complex plane.

only to the codons, but also to the features of the amino acids. Amino acids with similar properties (e.g., which cluster on state transition probability) tend to be neighbors in the complex representation of the genomic code in Figure 1.17.

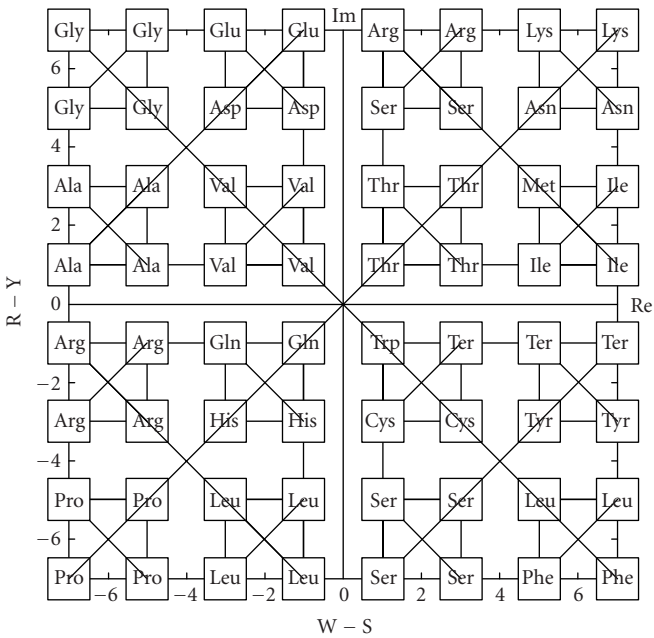


Figure 1.16. Mapping of the amino acids on the complex plane.

As mentioned above, it is possible to further reduce the dimensionality of the representation of nucleotide, codon, and amino acid sequences by using a real one-dimensional mapping. Table 1.3 gives the mapping of the digital codons to the numerical codes of the amino acids. The numerical values of the codons result from the base-four values of the nucleotides given in Table 1.2 and from the "nucleotide digits" in each codon. The numerical codes assigned to the amino acids result from the order of their first reference when gradually increasing the values of the codons from 0 to 63. By convention, the code zero is assigned to the terminator. As can be seen in the representations of the genetic code in Table 1.1 and in Figures 1.12, 1.13, 1.16, and 1.17, there are only two nondegenerated (one codon—one amino acid) mappings—for tryptophan and methionine, but nine double, one triple, five quadruple, and three sextuple degenerations, plus the three codons corresponding to the terminator. The minimum nonmonotonic dependency has only four reversals of the ascending order: for a terminator sequence and for the three instances of sextuple degeneration (leucine, serine, and arginine). An exhaustive search for all the 24 possible correspondences of the nucleotides to the digits 0–3 has shown that there does not exist a more monotonic mapping. The proposed mapping gives a piecewise constant function, with only the three mentioned reversals of the order, as shown in Table 1.3 and in Figure 1.18.

The reference to the various real and complex representations of the nucleotides can be simplified by using the pair of indices (p, q) as described in details in [23]. The index p specifies the *nucleotide permutations* and takes values from

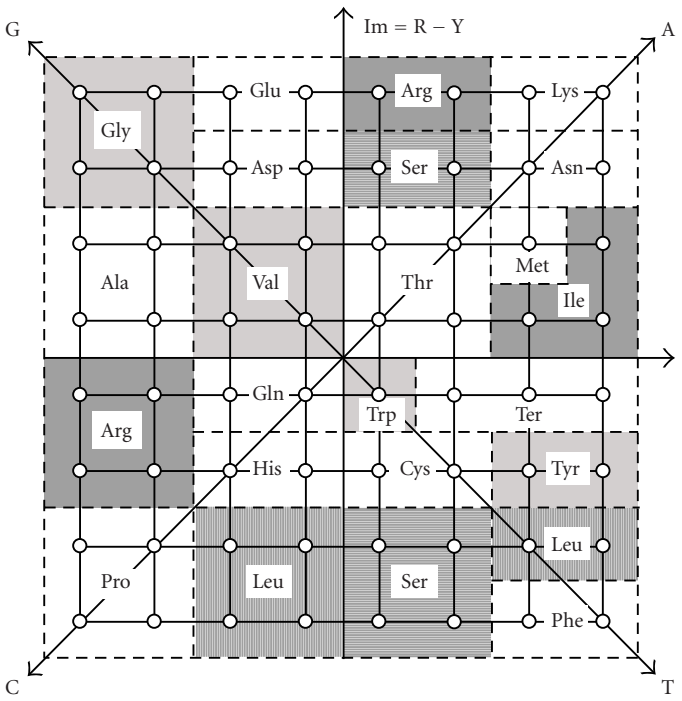


Figure 1.17. Genetic code complex representation.

1 to 24. The index q is used to specify the *representation type* and has the values: $q = 0$ for the real representation, $q = 1$ for a representation defined by the mapping of the nucleotides to pure real/pure imaginary numbers, and $q = 2$ for the mapping of nucleotides to quadrantally symmetric complex numbers, as defined by equation (1.4) and Figure 1.8 (for $p = 1$).

Despite the fact that the real representations of nucleotides described above are also one-to-one mappings, having an exact inverse, thus conserving the whole information in the initial symbolic sequence, the vectorial or complex representations are better fitted to reveal the basic features of the genomic sequences by their emphasis on the classes of nucleotides. Unfortunately, the simpler-to-handle real representations seem to be biased, as they induce some additivity of the properties of the mathematical representation, which does not have a direct correspondence in the nucleotide biochemical properties. In the following sections, we will present results obtained by using the complex (2D) and vectorial (3D) representations.

Complex representations have the advantage of expressing some of the biochemical features of the nucleotides in mathematical properties of their representations. For instance, the complementarity of the pairs of bases A−T, G−C is expressed by the fact that their representations are complex conjugates, while purines and pyrimidines have the same imaginary parts and opposite sign real parts. As already discussed, the complex representation of the codons and the amino acids

Table 1.3. Optimal correspondence of real numerical codons to amino acids.

Digital codon	Amino acid code	Long name	Short name	Symbol
10, 11, 14	0	Terminator	Ter	[end]
0, 1	1	Phenylalanine	Phe	[F]
2, 3, 16, 17, 18, 19	2	Leucine	Leu	[L]
4, 5, 6, 7, 44, 45	3	Serine	Ser	[S]
8, 9	4	Tyrosine	Tyr	[Y]
12, 13	5	Cysteine	Cys	[C]
15	6	Tryptophan	Trp	[W]
20, 21, 22, 23	7	Proline	Pro	[P]
24, 25	8	Histidine	His	[H]
26, 27	9	Glutamine	Gln	[Q]
28, 29, 30, 31, 46, 47	10	Arginine	Arg	[R]
32, 33, 34	11	Isoleucine	Ile	[I]
35	12	Methionine	Met	[M]
36, 37, 38, 39	13	Thereonine	Thr	[T]
40, 41	14	Asparagine	Asn	[N]
42, 43	15	Lysine	Lys	[K]
48, 49, 50, 51	16	Valine	Val	[V]
52, 53, 54, 55	17	Alanine	Ala	[A]
56, 57	18	Aspartic acid	Asp	[D]
58, 59	19	Glutamic Acid	Glu	[E]
60, 61, 62, 63	20	Glycine	Gly	[G]

shown in Figures 1.15 and 1.16 results simply from the projection of the codon and amino acid tetrahedrons in Figures 1.11 and 1.12 on the xOz plane. This leads naturally to the complex representation of the genetic code in Figure 1.17 and allows representing DNA sequences by complex signals at the levels of nucleotides, codons, and amino acids. It can be noticed that this complex mapping conserves the meaning of the distances between the codons, as resulting from the genetic code. Specifically, codons corresponding to the same amino acid are clustered in contiguous regions of the complex plane. From the frequency of the amino acids in the proteins, it results that the genetic code has some of the characteristics of Huffman (entropy) coding. Higher redundancy (degeneracy) in the encoding could correspond to primitive, older amino acids, while low redundancy, meaning a higher local resolution of the genetic code, could correspond to more recent amino acids. This hypothesis allows building models of ancestral proteins in the early times before the freezing of the genomic code.

Complex values can be attached in various ways to the amino acids. One modality is to assign to a certain amino acid the average value over the whole area onto which it is mapped, taking into account the relative frequencies of occurrence of the different codons that correspond to the amino acid. It has been shown that the assigning of the complex values to the nucleotides and to the amino acids can be adapted to various tasks. For instance, the optimum values for detecting the exons are different from the optimum ones for detecting the reading frames [35]. This gives the flexibility needed for targeting the approach to each application.

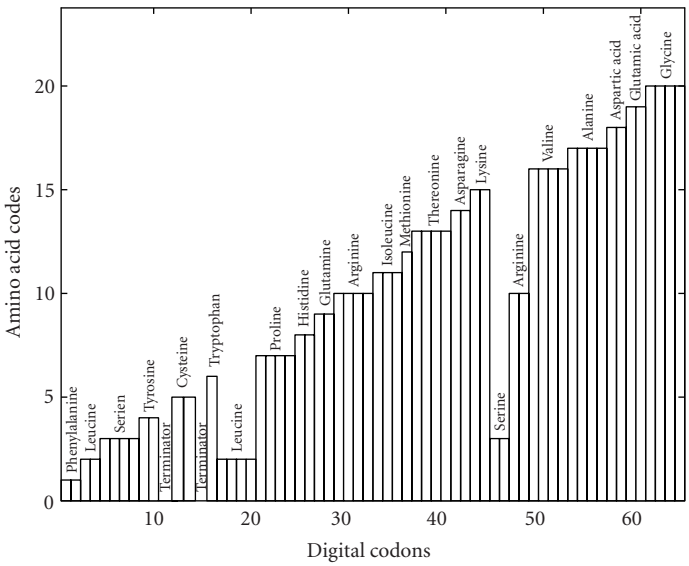


Figure 1.18. Optimal (minimally nonmonotonic) correspondence of numerical codons to amino acid codes.

For the analysis of large scale DNA features, only the nucleotide to complex mapping given in equations (1.4) and (1.5) and Figures 1.8 and 1.9 will be used.

1.4. Phase analysis of DNA sequences

All available complete genomes or available sets of contigs for eukaryote and prokaryote taxa have been downloaded from the GenBank [2] database of NIH, converted into genomic signals by using the mapping given in equation (1.4). The signals have been analyzed focussing on the extraction of large scale features of DNA sequences, up to the scale of whole chromosomes. Such properties transcend the differences between the coding (exons) and noncoding (introns) regions of DNA sequences, and refer to properties and functions of the chromosomes as whole entities. Several tools have been developed for this type of analysis, some also useful for local feature extraction, and have been presented elsewhere [23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. This section is devoted to the phase analysis of the complex genomic signals, which revealed some interesting large scale DNA features that could be important for better understanding such functions of chromosomes like replication, transcription, and crossover.

1.4.1. Fundamentals of phase analysis

The *phase* of a complex number is a periodic magnitude: the complex number does not change when adding or subtracting any multiple of 2π to or from its phase. To remove the ambiguity, the standard mathematical convention restricts

the phase of a complex number to the domain $(-\pi, \pi]$ that covers only once all the possible orientations of the associated vector in the complex plane. For the genomic signals obtained by using the mapping defined in Figure 1.8 and in equation (1.4), the *phases* of the nucleotide representations can have only the values $\{-3\pi/4, \pi/4, \pi/4, 3\pi/4\}$ radians.

The *cumulated phase* is the sum of the phases of the complex numbers in a sequence from the first element in the sequence, up to the current element. For the complex representation (1.4), the cumulated phase at a certain location along a sequence of nucleotides has the value:

$$\theta_c = \frac{\pi}{4}[3(n_G - n_C) + (n_A - n_T)], \tag{1.9}$$

where n_A, n_C, n_G, and n_T are the numbers of adenine, cytosine, guanine, and thymine nucleotides in the sequence, from the first to the current location. Consequently, the slope s_c of the cumulated phase along the DNA strand at a certain location is linked to the frequencies of occurrence of the nucleotides around that location by the equation:

$$s_c = \frac{\pi}{4}[3(f_G - f_C) + (f_A - f_T)], \tag{1.10}$$

where f_A, f_C, f_G, and f_T are the nucleotide occurrence frequencies.

The *unwrapped phase* is the corrected phase of the elements in a sequence of complex numbers, in which the absolute value of the difference between the phase of each element in the sequence and the phase of its preceding element is kept smaller than π by adding or subtracting an appropriate multiple of 2π to or from the phase of the current element. The unwrapped phase eliminates the phase jumps introduced by the conventional restriction of the phase domain described above and allows observing the true global phase trends along a sequence. For the complex representation given in equation (1.4), the *positive transitions* A → G, G → C, C → T, T → A determine an increase of the unwrapped phase, corresponding to a rotation in the trigonometric sense by $\pi/2$, the *negative transitions* A → T, T → C, C → G, G → A determine a decrease, corresponding to a clockwise rotation by $-\pi/2$, while all other transitions are *neutral*. A distinction has to be made between the exactly (first type) neutral transitions A ↔ A, C ↔ C, G ↔ G, T ↔ T, for which the difference of phase is zero in each instance, so that the unwrapped phase does not change, and the "on average" (second type) neutral transitions A → C, C → A, G → T, T → G, for which the difference of phase is $\pm\pi$. Because of the bias introduced by the conventional restriction of the phase to the domain $(-\pi, \pi]$, which favors π over $-\pi$, the standard unwrapped phase function and the corresponding functions implemented in most commercial software mathematics libraries, which apply the basic convention for the phase mentioned above, attach $+\pi$ to all the "on average" neutral transitions. This would distort the unwrapped phase and even the cumulated phase, if using complex representations that include real negative numbers (which is not the case for equations (1.4)). To avoid this unwanted effect, two solutions have been used:

(1) for large genomic sequences, from millions to hundreds of millions of nucleotides, uniformly distributed small random complex numbers have been added to each nucleotide complex representation, so that phases and differences of phase close to $-\pi$ are equally probable with the phases close to π and the artificial drift of the unwrapped phase towards positive values has been eliminated, (2) primarily for medium or small sequences, for example, when studying virus genomes, but also for large and very large sequences, a custom unwrapped phase function has been used that attaches zero phase change for all neutral transitions.

The accuracy of both procedures has been thoroughly verified using artificial sequences. It has been found that any bias related to the conventional restriction of the phase domain, which could affect crisp data processed with the standard unwrapped phase function, has been eliminated.

For the complex representation (1.4), taking the precautions mentioned above, the unwrapped phase at a certain location along a sequence of nucleotides has the value:

$$\theta_u = \frac{\pi}{2}(n_+ - n_-), \tag{1.11}$$

where n_+ and n_- are the numbers of the positive and negative transitions, respectively. The slope s_u of the variation of the unwrapped phase along a DNA strand is given by the relation:

$$s_u = \frac{\pi}{2}(f_+ - f_-), \tag{1.12}$$

where f_+ and f_- are the frequencies of the positive and negative transitions.

An almost constant slope of the unwrapped phase corresponds to an almost helicoidal wrapping of the complex representations of the nucleotides along the DNA strand. The step of the helix, that is, the spatial period over which the helix completes a turn, is given by

$$L = \frac{2\pi}{s_u}. \tag{1.13}$$

As will be shown in the next subsection, such an almost linear increase of the unwrapped phase, corresponding to a counter clockwise helix, is a long-range feature of all chromosomes of *Homo sapiens*, *Mus musculus*, and of other animal eukaryotes, while an opposite winding is common in plants and prokaryotes. The trend is maintained over distances of tens of millions of bases and reveals a regularity of the second-order statistics of the distribution of the succession of the bases which is a new property, distinct of Chargaff's laws.

It must be noted that the cumulated phase is related to the statistics of the nucleotides, while the unwrapped phase id related to the statistics of the pairs of nucleotides. Thus, the phase analysis of complex genomic signals is able to reveal features of both the first- and the second-order statistics of nucleotide distributions along DNA strands.

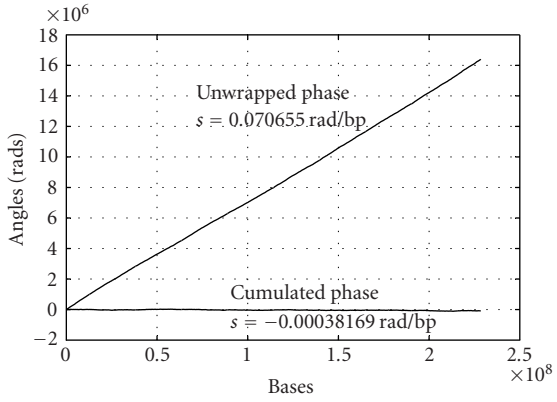


Figure 1.19. Cumulated and unwrapped phase along *Homo sapiens* chromosome 1 (phase 3, total length 228, 507, 674 bp [2]).

1.4.2. Phase analysis of eukaryote DNA sequences

Using the genomic signal approach, long-range features maintained over distances of 10^6–10^8 of base pairs, that is, at the scale of whole chromosomes, have been found in all available eukaryote genomes [31, 32]. The most conspicuous feature is an almost linear variation of the unwrapped phase found in all the investigated genomes, for both eukaryotes and prokaryotes. The slope is specific for various taxa and chromosomes.

Figure 1.19 presents the cumulated and unwrapped phase along concatenated phase-3 data for chromosome 1 of *Homo sapiens,* downloaded from GenBank [2]. Two main features of these phases are readily noticeable.

(i) The cumulated phase remains close to zero, in accordance to the second Chargaff's law for the *distribution* nucleotides—a first-order statistics, stating that the frequency of occurrence of purines and pyrimidines along eukaryote DNA molecules tend to be equal and balance each other [33].

(ii) The unwrapped phase has an almost linear variation maintained for the entire chromosome, for more than 228 millions of nucleotides, including both coding and noncoding regions. Such a behavior proves a rule similar to Chargaff's rule, but reveals a statistical regularity in the *succession* of the nucleotides—a second-order statistics, but reveals a statistical regularity in the *succession* of the nucleotides—a second-order statistics: *the difference between the frequencies of positive nucleotide-to-nucleotide transitions (*A → G, G → C, C → T, T → A*) and of negative transitions (the opposite ones) along a strand of nucleic acid tends to be small, constant, and taxon- and chromosome-specific* [28].

It is worth mentioning that less precise data tend to conform less to this rule, as can be seen from Figure 1.20 that presents the same plots as in Figure 1.19, but for all the concatenated contigs of chromosome 1 of *Homo sapiens*, comprising all the available 238,329,632 nucleotides, without filtering. As a practical use of the unwrapped phase quasilinearity rule, the compliance of a certain contig with the

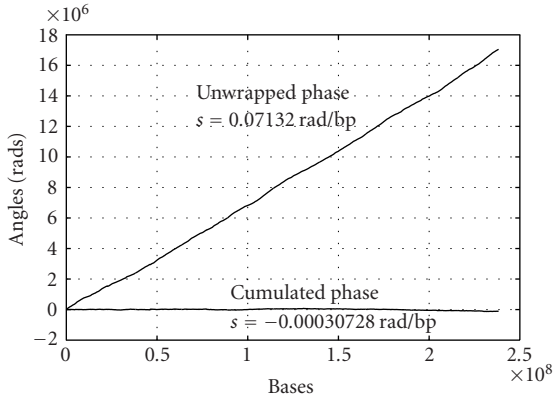


Figure 1.20. Cumulated and unwrapped phase along all concatenated contigs of *Homo sapiens* chromosome 1 (nonfiltered data, total length 238, 329, 632 bp [2, 3, 4]).

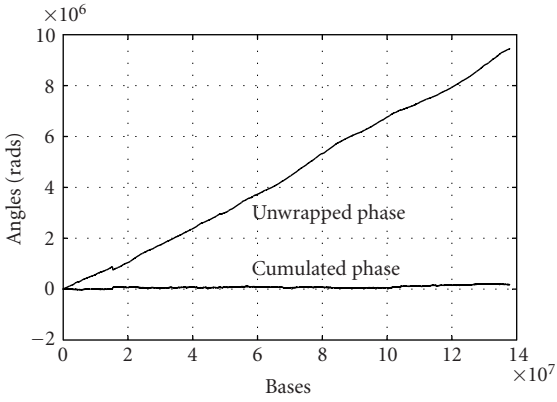


Figure 1.21. Cumulated and unwrapped phase along concatenated contigs of *Homo sapiens* chromosome 11 (older release, nonfiltered data, total length *xxx* bp [2]).

large scale regularities of the chromosome to which it belongs can be a used to spot out exceptions and errors. Figure 1.21 shows the cumulated phase and unwrapped phase along the ensemble of all concatenated contigs of *Homo sapiens* chromosome 11. The average slope of the unwrapped phase is $s_u = 0.0667$ rad/bp, while the various contigs have slopes in the range between 0.047 rad/bp = 2.7 degree/bp and 0.120 rad/bp = 6.9 degree/bp. A striking exception is found in the interval ∼ 15.17–15.38 Mbp of the concatenated string of contigs and corresponds to the contig of accession NT 029410 [2] for which the nucleotide complex representation phases are shown in Figure 1.22. On a length of about 210 Kbp, the unwrapped phase decreases linearly with a sharp average slope $s_u = -0.65$ rad/bp $= -37.2$ degree/bp, which corresponds to a large negative difference in the frequencies of positive and negative transitions $\triangle f_{pm} = f_+ - f_- = -39.4\%/\text{bp}$ and

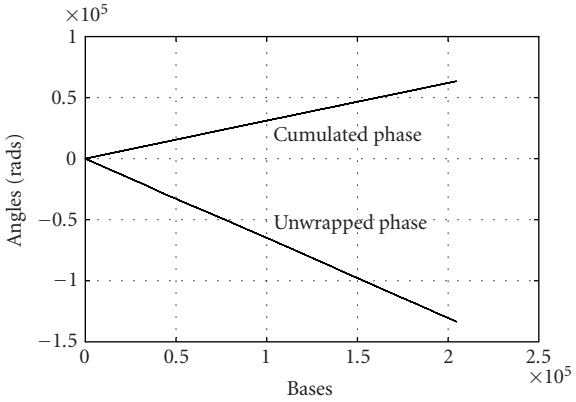


Figure 1.22. Cumulated and unwrapped phase along contig NT_029410 of *Homo sapiens* chromosome 11 (length *xxx* bp [2]).

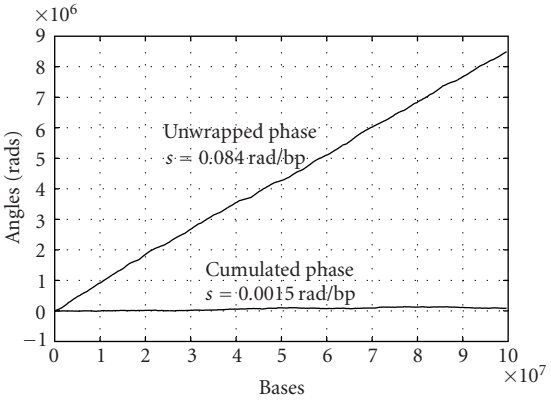


Figure 1.23. Cumulated and unwrapped phase for the available concatenated contigs of *Mus musculus* chromosome 11 (nonfiltered data, total length 99, 732, 879 bp [2, 5, 6]).

to a nucleotide average helix oriented clockwise, completing a turn for about every 9.7 bp. At the same time, the cumulated phase increases linearly with a slope $s_c = 0.325$ rad/bp = 18.6 degree/bp. This data seems to have been dropped from recent releases of chromosome 11 sequences.

Similar large scale properties can be found in all available eukaryote genomes. Figure 1.23 shows the phase diagram for the 99,732,879 nucleotides of the concatenated contigs of *Mus musculus* chromosome 11. The unwrapped phase increases also almost linearly with an average slope $s_u = 0.086$ rad/bp = 4.93 degree/bp, while the cumulated phase remains again almost constant at the scale of the diagram.

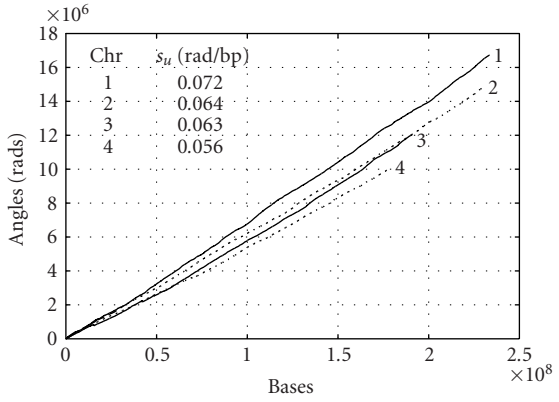


Figure 1.24. Unwrapped phase of the genomic signals for the nucleotide sequence of the concatenated contigs of *Homo sapiens* chromosomes 1–4 [2].

Such long-range regularities of the DNA molecules reveal a structuring of the genomic information at the level of whole chromosomes and contradict the assertion that genomes consist of scarce gene oases in an otherwise essentially empty, unstructured desert. Now it is accepted that the extragene regions can play significant functional roles at the level of the whole chromosome, in controlling processes like replication, transcription, crossover, and others. Along with many of the genes, the *Homo sapiens* and the *Mus musculus* genomes share twice as much other extragene DNA sequences. It is conjectured that these sequences must have important functions that explain how they were conserved over a divergent evolution of some 75 million years of the human and mouse lineages [1, 2, 3, 5, 7].

The approximately linear variation with positive slope has been found for the unwrapped phase of the genomic signals of all the chromosomes of *Homo sapiens* and *Mus musculus*. Figure 1.24 shows the results for the four largest chromosomes of *Homo sapiens*, while Figure 1.25 gives the curve for the shortest three chromosomes. Significant segments with negative slopes of the unwrapped phase have been found in *Homo sapiens* chromosomes 5, 8, 11, 17, 21, and Y. The average slope of the unwrapped phase is taxon and chromosome specific and has a functional role, most probably in controlling the movement Brownian machines like the DNA polymerase and in selecting homologous sites for the crossover exchange of genomic material. Table 1.4 shows the average slopes of the unwrapped phase for the concatenated contigs of *Homo sapiens* chromosomes currently available in the GenBank [2] data base.

1.4.3. Phase analysis of prokaryote DNA sequences

We start illustrating the phase features of prokaryote DNA sequences with the case of the well-studied *Escherichia coli*, for which the genome has been one of the first completely sequenced [14]. The most striking feature in Figure 1.26 is

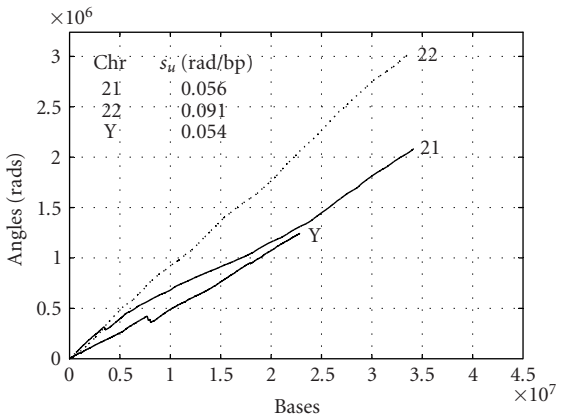


Figure 1.25. Unwrapped phase of the genomic signals for the nucleotide sequence of the concatenated contigs of *Homo sapiens* chromosomes 21, 22, Y [2].

Table 1.4. Average slopes of the unwrapped phase for the concatenated contigs of *homo sapiens* chromosomes.

Chr	s_u (rad/bp)	Chr	s_u (rad/bp)	Chr	s_u (rad/bp)	Chr	s_u (rad/bp)
1	0.072	7	0.066	13	0.057	19	0.084
2	0.064	8	0.062	14	0.066	20	0.073
3	0.063	9	0.066	15	0.072	21	0.057
4	0.056	10	0.067	16	0.075	22	0.091
5	0.060	11	0.069	17	0.078	X	0.057
6	0.062	12	0.068	18	0.060	Y	0.054

that the cumulated phase varies piecewise linearly along two domains of the circular DNA having almost equal length: a region of positive slope $s_{c+} = 0.0393$ rad/bp of length $l_+ = 2,266,409$ bp (split into two domains 1–1, 550, 413 bp and 3, 923, 226–4, 639, 221 bp) and a region of negative slope $s_{c-} = -0.0375$ rad/bp of length $l_- = 2,372,812$ bp. The quite sharp extremes of the cumulated phase are at 3, 923, 225 bp and 1, 550, 413 bp, respectively, very close to the experimentally found origin and terminus of chromosome replication. Quite similar diagrams have been obtained analyzing the difference in the occurrence frequencies of purines over pyrimidines R − Y = (A + G) − (T + C) (Figure 1.27) and of ketones over amines K − M = (G + T) − (C + A) (Figure 1.28). Figure 1.29 shows the excess of weak over strong bonds along the *Escherichia coli* DNA strand. As is well known, for prokaryotes most of the chromosome comprises encoding regions and in which cytosine and guanine are in excess over adenine and thymine.

It is rather surprising that the variation closest to (piecewise) linear is found for the cumulated phase, which has a slope dependent on a mixture of the nucleotide occurrence frequencies given by equation (1.10). Again, the variation of the unwrapped phase is almost linear for the whole chromosome (Figure 1.30) and passes without change over the points where the slope of the cumulated phase

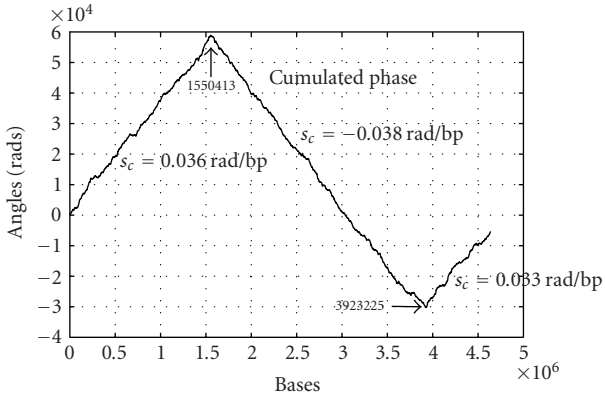


Figure 1.26. Cumulated phase for the circular chromosome of *Escherichia coli* K12 (NC_000913, complete genome, length 4, 639, 221 bp [2, 14]).

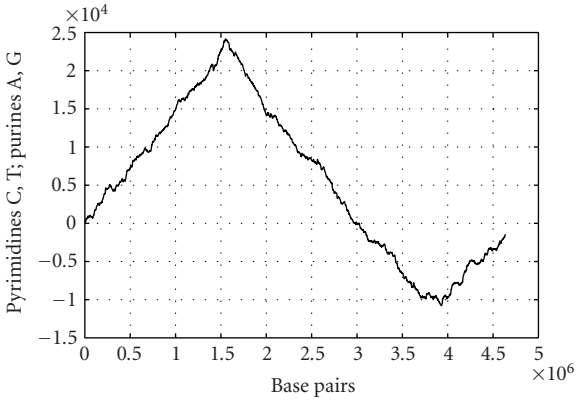


Figure 1.27. Purine over pyrimidine excess (A + G) − (T + C) along the circular chromosome of *Escherichia coli* K12 (NC_000913, complete genome, length 4, 639, 221 bp [2, 14]).

changes sign. This is a general feature, found for all chromosomes and all prokaryotes, and will be discussed in the next section of this chapter.

Figure 1.31 shows the cumulated and the unwrapped phase along the circular chromosome of *Yersinia pestis* [18] (accession number NC_003143 [2]). As in the case of *Escherichia coli*, the breaking points are most probably in relation with the origins and the termini of chromosome replichores, but we are not aware of the corresponding experimental results. It is to be noticed that, in opposition to *Escherichia coli* [14] and *Bacillus subtilis* [16] which display only one maximum and one minimum [24], the cumulated phase of *Yersinia pestis* shows four points of each type. This corresponds to the fusion of more strains into the circular chromosome of *Yersinia pestis* and could reveal aspects of the ancestral history of the pathogen. The change of sign of the cumulated phase slope at the breaking points shows that there is a cut and a macroswitch of the two DNA strands, so that the

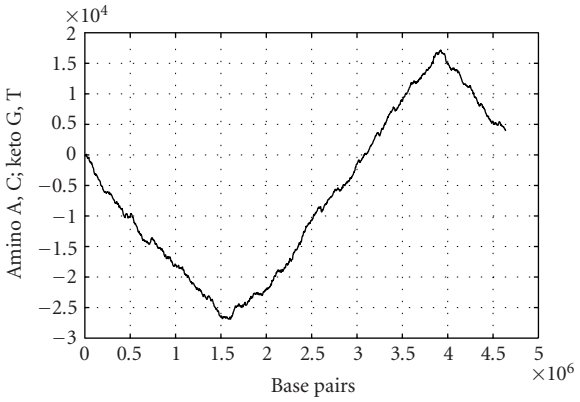


Figure 1.28. Keto over amino excess (G + T) − (C + A) along the chromosome of *Escherichia coli* (NC_000913 [2, 14]).

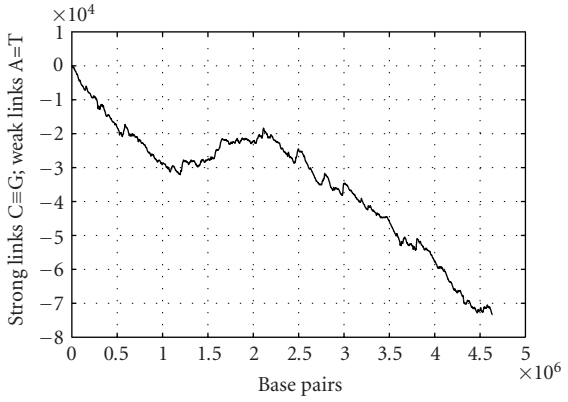


Figure 1.29. Weak bonds over strong bonds W − S = (A + T) − (C + C) along the chromosome of *Escherichia coli* (NC_000913 [2, 14]).

difference between the frequencies of occurrence of the nucleotides changes the sign. It is remarkable that, in the same points, there is little or no change in the unwrapped phase. This will be explained in the next section of this chapter based on a longitudinal model of the chromosomes' "patchy" structure.

Similar characteristics have been found for almost all other studied prokaryotes. Figure 1.32 presents the cumulated and unwrapped phase for an intracellular pathogen of humans: *Chlamydophila pneumoniae* CWL029 (NC_000922 [34]). Again the linear regions correspond to the "replichores" of bacterial circular chromosomes, and the extremes of the cumulated phase are the origin and terminus of chromosome replication. The differences in nucleotide occurrence frequencies have been explained by the differences in mutation probabilities resulting from the asymmetry of replication mechanisms for the leading and lagging strands but, most probably, this statistically ordered nonhomogenity plays a fundamental

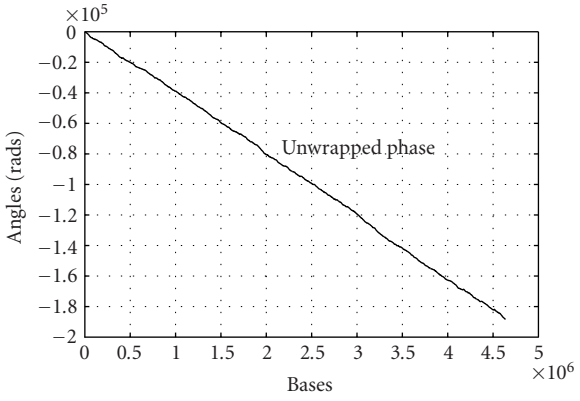


Figure 1.30. Unwrapped phase for the circular chromosome of *Escherichia coli* (NC_000913 [2, 14]).

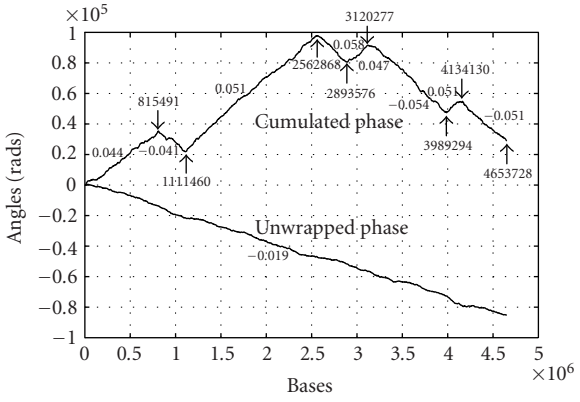


Figure 1.31. Unwrapped and cumulated phase for the circular chromosome of *Yersinia pestis* (NC_0003143, complete genome, length 4, 653, 728 bp [2, 18]).

role in the functioning of some "molecular machines," like DNA polymerase that moves along a DNA strand by converting the thermal motion in an ordered displacement.

It has been shown recently that DNA molecules have a fractal-like structure resulting from their long-range correlations [30]. The self-similarity, that is, the fractal-like structure is revealed by the linearity of the plot $\log(N)$ versus $\log(B)$, where N is the number of filled boxes of size B, while the slope gives the fractal dimension. From the analysis of the cumulated phase of the circular chromosome of *Chlamydophila pneumoniae* CWL029 in Figure 1.32, with a 1024 bp sliding window, an average fractal dimension of 1.05 has been found, only slightly higher than one, in accordance with the long correlations observed in the cumulated and unwrapped phase curves.

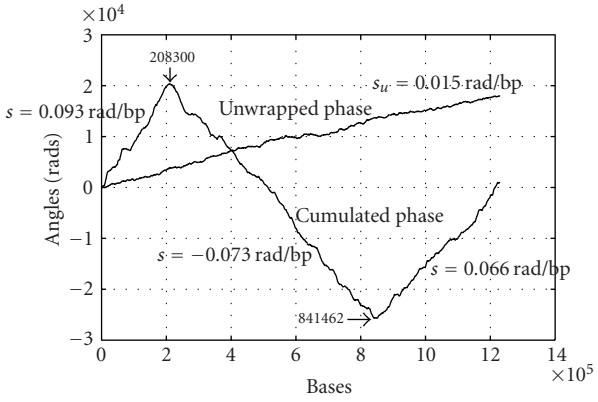


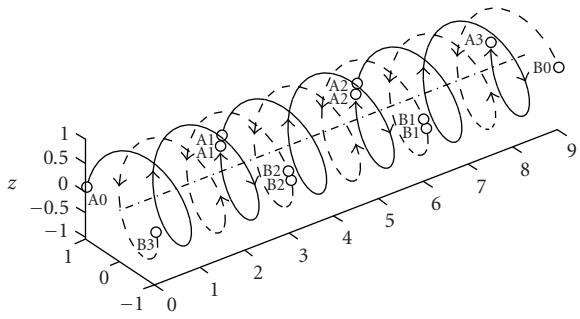
Figure 1.32. Cumulated and unwrapped phase for the circular chromosome of *Chlamydophila pneumoniae* CWL029 (NC_000922, complete genome, length 1,230,230 bp [2]).

1.5. Phase analysis of reoriented ORFs

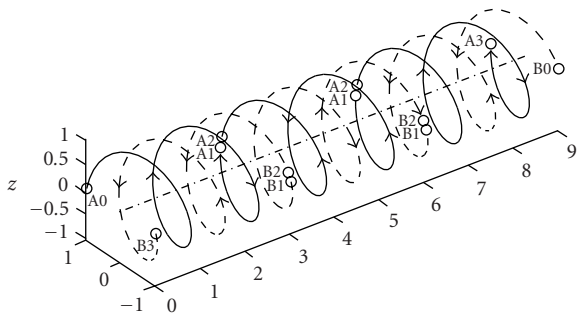
As discussed in Section 1.2, each DNA strand has a well-defined positive direction (the 5′ → 3′ sense), along which successive nucleotides can be joined to each other [29]. The two strands of a DNA double helix have opposite positive directions. DNA molecules have a very "patchy" structure with intertwined coding and noncoding segments oriented in both direct and inverse sense [36]. For most currently sequenced genomes, the information about the direct or inverse orientation of the coding regions—the ORF—has been identified and is available in the genomic databases [2].

The main point that results from the analysis of the modalities in which DNA segments can be chained together along a DNA double helix is that a direction reversal of a DNA segment is always accompanied by a switching of the antiparallel strands of its double helix. This property is a direct result of the requirement that all the nucleotides be linked to each other along the DNA strands only in the 5′ to 3′ sense.

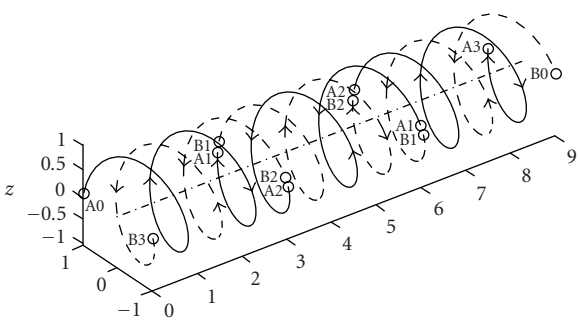
Figure 1.33 schematically shows the way in which the positive orientation restriction is satisfied when a segment of a DNA double helix is reversed and has simultaneously switched its strands. In Figure 1.33a, the chains $(A_0A_1)(A_1A_2)(A_2A_3)$ and $(B_0B_1)(B_1B_2)(B_2B_3)$ have been marked on the two strands, having the positive (5′ to 3′) directions as indicated by the arrows. The reversal of the middle segment, without the corresponding switching of its strands (Figure 1.33b), would generate the forbidden chains $(A_0A_1)(A_2A_1)(A_2A_3)$ and $(B_0B_1)(B_2B_1)(B_2B_3)$ that violate the 5′ to 3′ alignment condition. Similarly, the switching of the strands of the middle segment, without its reversal, would generate the equally forbidden chains $(A_0A_1)(B_2B_1)(A_2A_3)$ and $(B_0B_1)(A_2A_1)(B_2B_3)$, not shown in Figure 1.33. Finally, only the conjoint reversal of the middle segment and the switching of its strands (Figure 1.33c) generate the chains $(A_0A_1)(B_1B_2)(A_2A_3)$ and $(B_0B_1)(A_1A_2)(B_2B_3)$, which are compatible with the 5′ to 3′ orientation condition.



(a)



(b)



(c)

Figure 1.33. Schematic representation of a DNA segment direction reversal: (a) the two antiparallel strands have the segments ordered in the 5′ to 3′ direction indicated by arrows; (b) hypothetic reversal of the middle segment, without the switching of the strands; (c) direction reversal and strand switching for the middle segment. The 5′ to 3′ alignment condition is violated in case (b) but reestablished for (c).

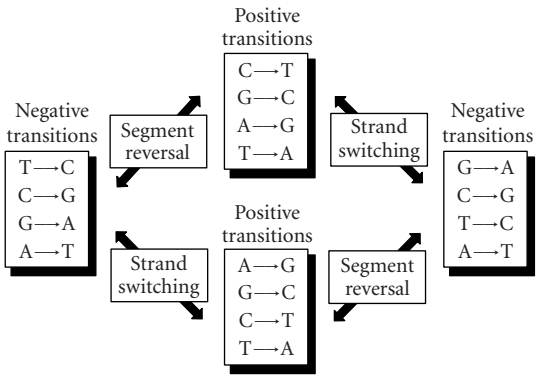


Figure 1.34. Interchange of positive and negative nucleotide-to-nucleotide transitions after *segment reversal and strand switching.*

We also mention that, in order to practically perform such a reversal, the two branches (A_1A_2) and (B_1B_2) of the DNA double helix segment should not be exactly aligned, but slightly shifted with respect to each other and the "free" nucleotides at the two ends should be complementary, to provide the necessary "sticky ends" allowing the easy reattachment of the strands. This condition does not affect the aspects discussed here.

As a consequence of the coupling of the direction reversal with the strand switching imposed by the condition to maintain the continuity of the positive directions ($5' \rightarrow 3'$) along the two strands of the DNA molecule, there is always a pair of changes when a DNA segment is inversely inserted. Thus, the sense/antisense orientation of individual DNA segments affects only the nucleotide frequencies, but conserves the frequencies of the positive and negative transitions. Figure 1.34 shows how the type of nucleotide-to-nucleotide transitions changes (positive to negative, and *vice versa*) for a segment reversal and for a strand switching. The reversal of an individual DNA segment affects only the first-order statistics of the nucleotides, while the second-order statistics remains unchanged. Thus, the cumulated phase of a genomic signal, which depends on the frequency of nucleotides along the corresponding DNA strand, changes significantly for a segment reversal, while the unwrapped phase, which depends on second-order statistical features, does not.

This model explains why the unwrapped phase has a regular, almost linear, variation even for eukaryote chromosomes [23, 24], despite their very high fragmentation and quasirandom distribution of direct and inverse DNA segments, while the cumulated phase has only a slight drift close to zero.

Figure 1.35 shows together the cumulated phase and the unwrapped phase of the genomic signal for the complete circular chromosome of *Escherichia coli* [14] (NC_000913 [2]) comprising 4,639,221 bp (also shown in Figures 1.26 and 1.30) and for the 4,290 concatenated reoriented coding regions, comprising 4,097,248 bp. All the coding regions having an inverse reading frame have been inversed

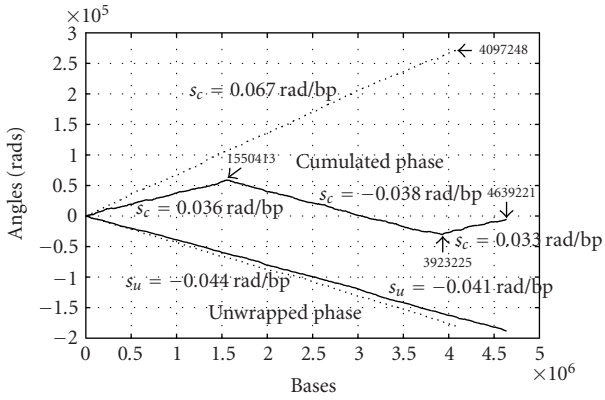


Figure 1.35. Cumulated and unwrapped phase of the genomic signals for the complete genome (4, 639, 221 bp) and the 4, 290 concatenated reoriented coding regions (4, 097, 248 bp) of *Escherichia coli* (NC_000913 [2, 14]).

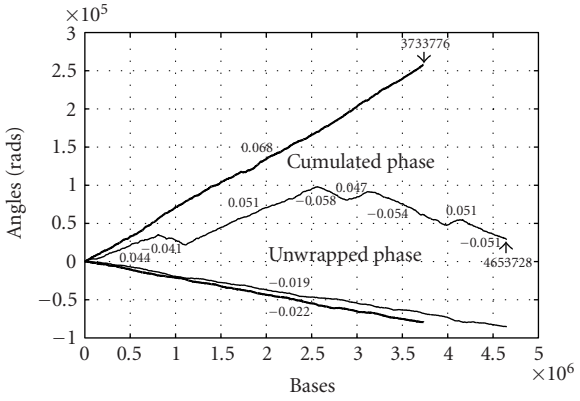


Figure 1.36. Cumulated and unwrapped phase of the genomic signals for the complete genome (4, 653, 728 bp) and the 4034 concatenated reoriented coding regions (3, 733, 776 bp) of *Yersinia pestis*[3,17] (accession number NC_003143 [2, 18]).

and complemented (i.e., A and T, on one hand, C and G, on the other, have been interchanged to account for strand switching). The disappearance of the breaking points in the cumulated phase under the effect of the reorienting is evident, while the unwrapped phase changes little.

Similarly, Figure 1.36 shows the cumulated and the unwrapped phase for the complete circular chromosome of *Yersinia pestis* strain CO92 (accession number NC_003143) with a length of 4, 853, 728 bp and for its concatenated reoriented 3, 884 coding regions comprising 3, 733, 776 bp. The slope of the cumulated and the unwrapped phases are changed not only because the intergene regions have been eliminated, but also because direct and inverse coding regions are actually

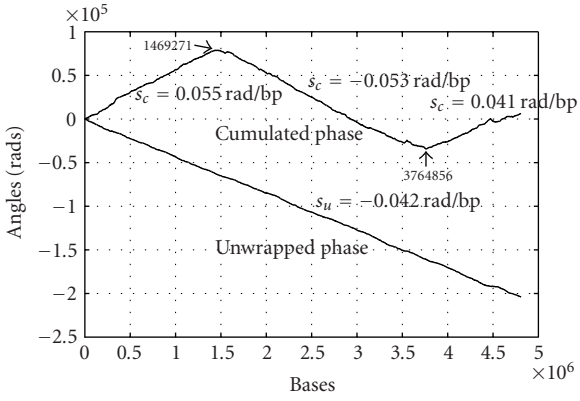


Figure 1.37. Cumulated and unwrapped phase for the circular chromosome of *Salmonella typhi* (AL_5113382, length 4, 809, 037 bp [2]).

distributed in all the four positive and four negative slope segments of the cumulated phase, certainly, with very different frequencies. The orientation of the coding regions correlates well with the slope of the cumulated phase: most direct ORF are in the positive slope regions, while most inverse ORF are in the negative slope regions.

Figure 1.37 presents the cumulated and the unwrapped phase of the complete circular chromosome *Salmonella typhi*, the multiple drug resistant strain CT18 (accession AL_513382 [2]). The locations of the breaking points, where the cumulated phase changes the sign of the slope of its variation along the DNA strand, are given in the figure. Even if locally the cumulated phase and the unwrapped phase have not a smooth variation, at the scale used in Figure 1.37, the variation is quite smooth and regular. A pixel in the lines in Figure 1.37 represents 6050 data points, but the absolute value of the difference between the maximum and minimum values of the data in the set of points represented by each pixel is smaller than the vertical pixel dimension expressed in data units. This means that the local data variation falls between the limits of the width of the line used for the plot, so that the graphic representation of data by a line is fully adequate. The conditions for signals graphical representability as lines will be presented in more detail in the next section of this chapter. As shown in the previous section for other prokaryotes, the cumulated phase has an approximately piecewise linear variation over two almost equal domains, one of positive slope (apparently divided in the intervals 1–1469271 and 3764857–4809037, but actually contiguous on the circular chromosome) and the second of negative slope (1469272–3764856), while the unwrapped phase has an almost linear variation for the entire chromosome, showing little or no change in the breaking points. The breaking points, like the extremes of the integrated skew diagrams, have been put in relation with the origins and termini of chromosome replichors [28, 37, 38]. The slope of the cumulated phase in each domain is related to the nucleotide frequency in that domain by equation

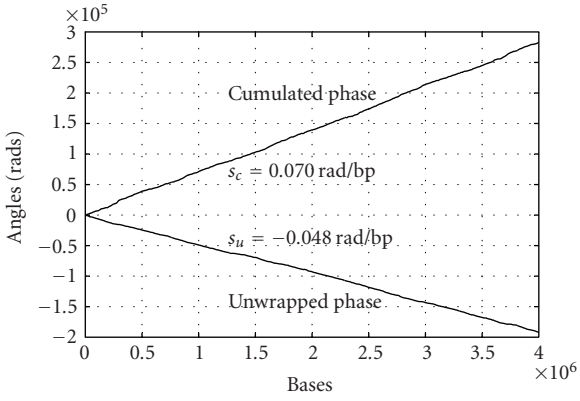


Figure 1.38. Cumulated and unwrapped phase of the concatenated 4393 reoriented coding regions (3,999,478 bp) of *Salmonella typhi* genome (AL_5113382 [2]).

(1.10). In the breaking points, apparently a macroswitching of the strands, accompanied by the reversal of one of the domain-large segments, occurs. The two domains comprise a large number of much smaller segments, oriented in the direct and the inverse sense. At the junctions of these segments, the reversal and switching of DNA helix segments, as described in the previous section, take place. The average slope of each large domain is actually determined by the density of direct and inverse small segments along that domain. Because the intergenic regions, for which the orientation is not known, have to be left out of the reoriented sequence, the new sequence is shorter than the one that contains the entire chromosome or all the available contigs given in the GenBank data base [2].

Figure 1.38 shows the cumulated and unwrapped phase of the genomic signal obtained by concatenating the 4393 reoriented coding regions of *Salmonella typhi* genome (accession AL_5113382 [2]). Each inverse coding region (inverse ORF) has been reversed and complemented, that is, the nucleotides inside the same W (adenine-thymine) or S (cytosine-guanine) class have been replaced with each other, to take into account the switching of the strands that accompanies the segment reversal. As expected from the model, the breaking points in the cumulated phase disappear and the absolute values of the slopes increase, as there is no longer interweaving of direct and inverse ORFs. The average slope s_c of the cumulated phase of a genomic signal for a domain is linked to the average slope $s_c^{(0)}$ of the concatenated reoriented coding regions by the relation:

$$s_c = \frac{\sum_{k=1}^{n_+} l_k^{(+)} - \sum_{k=1}^{n_-} l_k^{(-)}}{\sum_{k=1}^{n_+} l_k^{(+)} + \sum_{k=1}^{n_-} l_k^{(-)}} s_c^{(0)}, \tag{1.14}$$

where $\sum_{k=1}^{n_+} l_k^{(+)}$ and $\sum_{k=1}^{n_-} l_k^{(-)}$ are the total lengths of the n_+ direct and n_- inverse ORFs in the given domain.

The unwrapped phase, which is linked by equation (1.12) to the nucleotide positive and negative transition frequencies, shows little or no change when replacing the chromosome nucleotide sequence with the concatenated sequence of reoriented coding regions. As explained, the reorientation of the inverse coding regions consists in their reversal and switching of their strands. Figure 1.34 shows the effect of the segment reversal and strand switching transformations on the positive and negative nucleotide-to-nucleotide transitions for the case of the complex genomic signal representation given by equation (1.1). After an even number of segment reversal and/or strand switching transformations of a DNA segment, the nucleotide transitions do not change their type (positive or negative). As a consequence, the slope of the unwrapped phase does not change.

It is remarkable that the approximately piecewise linear variation of the cumulated phase for the whole chromosome, comprising two complementary regions—also found by skew diagrams techniques [36, 38]—is replaced with an approximately linear variation over the whole sequence, when reorienting all coding regions in the same reference direction. This result could suggest the existence of an ancestral chromosome structure with a single global statistical regularity, which has evolved into a more complex structure by the reversal of the direction for a significant part of DNA segments.

Similar results have been found in the phase analysis of many other genomic signals corresponding to circular and linear chromosomes of various prokaryotes. A special case is the aerobic hyperthermophilic crenarchaeon *Aeropyrum pernix K*, for which the genome comprising $1,669,695$ base pairs has been completely sequenced [3]. The unwrapped phase varies almost linearly, in agreement with the similar results found for all the other investigated prokaryote and eukaryote genomes, confirming the rule stated in the previous section. But the cumulated phase decreases irregularly, an untypical behavior for prokaryotes that tend to have a regular piecewise linear variation of the cumulated phase along their circular DNA molecules, as shown above. Nevertheless, the cumulated phase of the $1,553,043$ base pairs signal corresponding to the sequence obtained by concatenating the $1,839$ coding regions, after reorienting them in the same reference direction, becomes approximately linear, while the unwrapped phase remains unchanged.

We conjecture that the fine combining of DNA segments with opposite orientation, in order to generate certain well-defined values of the slope of the cumulated phase, that is, certain densities of the repartition of nucleotides, has a functional role at the level of the chromosomes, most probably in processes like replication, transcription, or cross over. The particular statistical structure of DNA molecules that generates this specific shape of the cumulated and unwrapped phases could play an important role in the mutual recognition and alignment of interacting regions of chromosomes and the separation of the species. The first- and second-order statistical regularities, resulting from the specific variation of the unwrapped and cumulated phases, can be put in correspondence with the molecule potentials produced by available hydrogen bonds and can be used to describe the interaction of a given DNA segment with proteins and with other

DNA segments in processes like replication, transcription, or crossover. An example is the movement of DNA polymerase along a DNA strand, operating like a "Brownian machine that converts random molecular movements into an ordered gradual advance during replication. The speed of movement can be expressed as a function of the temperature and the slope of the phase. These hypotheses are also sustained by the fact that the emergence of an almost linear variation of the cumulated phase after the reorientation of all coding regions is a property found in both circular and linear chromosomes of prokaryotes, but not in the plasmids.

1.6. Representability of genomic signals

1.6.1. Well-fitted screens and the data scattering ratio

When operating with large sets of data, especially data describing complex systems or processes or generated by such systems or processes, with a possibly chaotic or random dynamics, the problem of adequate representation of data is central. The final understanding of any set of data or signals lays with human operators for which the graphical representation, allowing to grasp at once features hidden in piles of numerical data, is the natural choice, at least as a guide to the precise values. As shown in the previous sections of this chapters, symbolic nucleotide sequences can be converted into digital genomic signals by using the complex (2D) quadrantal representation derived from the tetrahedral (3D) representation of nucleotides. The study of complex genomic signals, using signal processing methods, reveals large scale features of chromosomes that would be difficult to grasp by applying only the statistical or pattern matching methods currently used in the analysis of symbolic genomic data. In the context of operating with a large volume of data, at various resolutions, and visualizing the results to make them available to humans, the problem of data representability becomes critical. In the following, we present an analysis of data representability based on the concept of the data scattering ratio of a pixel. Representability diagrams are traced for several typical cases of standard signals and for some genomic signals. It is shown that the variation of genomic data along nucleotide sequences, specifically the cumulated and unwrapped phase, can be visualized adequately as simple graphic lines for low and large scales, while for medium scales (thousands to tens of thousands of base pairs) the statistical-like description must be used.

Figure 1.39 shows the plot as a line of the digital signal $s[i]$, $i \in I^S = \{1, \ldots, L\}$, where L is the length of the sequence or subsequence of data to be represented. One pixel is extracted and magnified to allow comparing the absolute value of the variation V_y of the signal for the set of samples represented by the pixel with the pixel height P_y measured in signal units. For the case in the figure, which corresponds to real data giving the unwrapped phase of the complete genome of *Bacillus subtilis*, we have $V_y < P_y$, so that the graphical representation of the data by a line with the width of a pixel is adequate in that point and, actually, for the whole sequence. The size of the screen in pixels is considered fixed, for example, the usual screen size $N_x = 1024$ by $N_y = 768$ pixels. To optimally use the screen to represent the

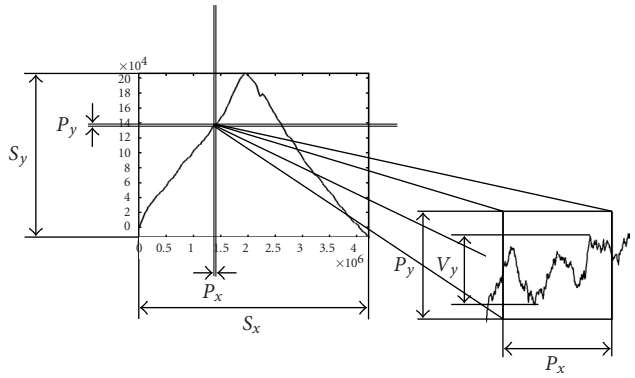


Figure 1.39. Data-fitted screen and a magnified pixel.

data, the available screen space must be fitted to the data: the horizontal screen size S_x, in number of samples, has to be made equal to the length L of the sequence (or subsequence) to be represented, while the screen vertical size S_y, in data units, must be chosen equal to the absolute value of the variation of the data in the represented sequence:

$$S_y = \max_{i \in I^S} \left(s[i]\right) - \min_{i \in I^S} \left(s[i]\right). \tag{1.15}$$

Correspondingly, the horizontal and vertical pixel sizes are given by

$$P_x = \frac{S_x}{N_x}, \qquad P_y = \frac{S_y}{N_y}, \tag{1.16}$$

in number of samples and data units, respectively.

The variation of the data for the set of samples corresponding to a pixel is

$$V_y(h) = \max_{i \in I_h^P} \left(s[i]\right) - \min_{i \in I_h^P} \left(s[i]\right), \tag{1.17}$$

where $I_h^P = \{(h-1)P_x + 1, \ldots, hP_x\}; h = 1, \ldots, N_x$.

As mentioned above, the adequateness of the representation of the set of P_x data samples by just one a pixel can be characterized by the ratio:

$$Q(h) = \frac{V_y(h)}{P_y} \tag{1.18}$$

that we will call the data scattering ratio of the pixel h.

If $Q \leq 1$, the pixel represents properly all the data samples it represents and covers. When all the pixels in a line satisfy this condition, the data can be represented adequately by a line having the width of one pixel. If Q is below two or

three units for every pixel of the sequence fitted in the screen, the data can also be represented properly by a line, but the width of the line must correspond to the maximum value of Q. When Q has larger values, but the data is densely and quite uniformly distributed, so that Q is approximately the same for all the pixels and there are no outliers, the data can be represented adequately by a couple of lines showing the maximum and minimum values of the data for each pixel. Finally, if the data is scattered and/or there are outliers, this approach is no longer practical and a statistical-like description of data is needed for their representation. The pixel can be considered a sliding window of size P_x. If the data distribution is close enough to a normal distribution, the data can be described for each such window by the mean value and the standard deviation. A line giving the mean value and two other lines or some error bars for delimiting some confidence interval expressed in terms of the standard deviation can be used to represent the data.

In the following, we analyze the representability of several types of data and signals, including genomic signals, in terms of their representability characteristic

$$\tilde{Q} = \frac{\tilde{V}_y}{P_y} = f(P_x), \tag{1.19}$$

where $\tilde{Q} = \tilde{V}_y/P_y$ is the average data scattering ratio for all the pixels in the represented line, with $\tilde{V}_y = \text{mean}_{h=1,\ldots,N_x}(V_y(h))$, while P_x is the pixel horizontal size. When drawing the representability diagram showing the representability characteristic (1.19), logarithmic scales in base 2 will be used for both abscissa and ordinate. Correspondingly, the pixel size $P_x^{(k)}$ will be increased in a geometrical scale with ratio two:

$$P_x^{(1)} = 1,\ldots, \qquad P_x^{(k)} = 2^{k-1},\ldots, \qquad P_x^{(k_{\max})} = 2^{k_{\max}-1}, \tag{1.20}$$

so that the screen horizontal size $S_x^{(k)} = N_x P_x^{(k)}$, $k = 1,\ldots,k_{\max}$, will also double at each step for a fixed N_x. The number of steps necessary to cover the whole sequence of length L is $k_{\max} = \lfloor \log_2 L/N_x \rfloor + 1$, where $\lfloor x \rfloor$ denotes the smaller integer larger than or equal to x. In this case, the largest screen equals or exceeds the length of the sequence. The number of screens necessary to represent the whole sequence at step k is

$$N_S^{(k)} = \left\lfloor \frac{L}{S_x^{(k)}} \right\rfloor = \left\lfloor \frac{L}{N_x 2^{k-1}} \right\rfloor. \tag{1.21}$$

If the length L of the sequence is not a power of two, the last screen at each step k, including the largest screen for the last step, might not be well fitted to the data and will be excluded from the diagram. When $L = 2^m$ and $N_x = 2^s$, all screens will be horizontally fitted to the data and their number $N_S^{(k)} = 2^{m-s+1-k} = 2^{k_{\max}-k}$ will form a geometrically decreasing sequence with ratio 1/2, from $2^{k_{\max}-1}$ to 1. Each screen (window) will be vertically fitted to the data, by choosing its vertical

size equal to the absolute value of the variation of the data in that screen:

$$S_y^{(k)}(j) = \max_{i \in I_j^S} (s[i]) - \min_{i \in I_j^S} (s[i]), \quad j = 1, \ldots, j_{\max}^{(k)}, \tag{1.22}$$

where $I_j^S = \{(j-1)S_x^{(k)} + 1, \ldots, jS_x^{(k)}\}$ are the indices of the samples represented in the screen j and $j_{\max}^{(k)} = N_S^{(k)}$ is the number of screens at step k.

A 3D diagram will be used to show the variation of the average data scattering ratio for the pixels in each of the screens used to cover all the length of the sequence L at various pixel sizes.

1.6.2. Representability best case: monotonic signals

In the case of monotonically increasing signals, the relation (1.13) for the vertical size of screen j becomes

$$S_y^{(k)}(j) = s\left[jS_x^{(k)}\right] - s\left[(j-1)S_x^{(k)} + 1\right], \tag{1.23}$$

so that the average screen height results:

$$\tilde{S}_y^{(k)} = \operatorname*{mean}_{j=1,\ldots,N_S^{(k)}} \left(S_y^{(k)}(j)\right) = \frac{1}{N_S^{(k)}} \sum_{j=1}^{N_S^{(k)}} \left(s\left[jS_x^{(k)}\right] - s\left[(j-1)S_x^{(k)} + 1\right]\right). \tag{1.24}$$

Using $j_{\max}^{(k)} S_x^{(k)} = L$, this expression can be rewritten as

$$\tilde{S}_y^{(k)} = \frac{2^{k-1}N_x}{L}\left(s[L] - s[1] - \sum_{j=1}^{N_S^{(k)}-1} \left(s\left[jS_x^{(k)} + 1\right] - s\left[jS_x^{(k)}\right]\right)\right), \tag{1.25}$$

where the sum contains signal variations between samples at distance one, subsampled with the step $S_x^{(k)}$. A similar expression holds for monotonically decreasing signals, so that the average screen height for monotonic signals results:

$$\tilde{S}_y^{(k)} = \frac{2^{k-1}N_x}{L}\left(s[L] - s[1] - \left(j_{\max}^{(k)} - 1\right) \operatorname{mean}\left(|d|\right)_{\downarrow S_x^{(k)}}\right), \tag{1.26}$$

where

$$\operatorname{mean}\left(|d|\right)_{\downarrow S_x^{(k)}} = \operatorname*{mean}_{j=1,\ldots,j_{\max}^{(k)}-1} \left(\left|d\left[jS_x^{(k)}\right]\right|\right) = \frac{1}{j_{\max}^{(k)} - 1} \sum_{j=1}^{j_{\max}^{(k)}-1} \left|d\left[jS_x^{(k)}\right]\right| \tag{1.27}$$

is the average absolute variation of the signal between samples at distance one, down-sampled at the step $S_x^{(k)}$.

Similarly, from (1.17) results the average variation of the data for sets of samples corresponding to pixels:

$$\tilde{V}_y^{(k)} = \frac{2^{k-1}}{L}\Big(s[L] - s[1] - (h_{\max}^{(k)} - 1)\,\text{mean}\,(|d|)_{\downarrow P_x^{(k)}}\Big), \tag{1.28}$$

where

$$\text{mean}\,(|d|)_{\downarrow P_x^{(k)}} = \underset{h=1,\ldots,h_{\max}^{(k)}-1}{\text{mean}}\,(|\,d[hP_x^{(k)}]\,|) = \frac{1}{h_{\max}^{(k)} - 1}\sum_{h=1}^{h_{\max}^{(k)}-1}|\,d[hP_x^{(k)}]\,| \tag{1.29}$$

is the average absolute variation of the signal between samples at distance one, down-sampled at the pixel step $P_x^{(k)}$.

As a consequence, the average data scattering ratio for a monotonic signal is given by the equation

$$\tilde{Q}^{(k)} = \frac{\tilde{V}_y^{(k)}}{\tilde{P}_y^{(k)}} = \frac{N_y}{N_x}\frac{s[L] - s[1] - (N_P^{(k)} - 1)\,\text{mean}\,(|d|)_{\downarrow P_x^{(k)}}}{s[L] - s[1] - (N_S^{(k)} - 1)\,\text{mean}\,(|d|)_{\downarrow S_x^{(k)}}}, \tag{1.30}$$

where $N_P^{(k)}$ is the total number of pixels to represent the sequence $s[i]$, $i = 1,\ldots,L$, for a horizontal pixel size $P_x^{(k)} = 2^{k-1}$, $N_S^{(k)} = N_P^{(k)}/N_x$ is the total number of screens necessary to represent the data at resolution k, and mean$(|d|_{\downarrow D})$ is the average of the absolute values of the signal variation between successive samples $d[i] = s[i+1] - s[i]$, down-sampled at step D. As long as the sampling density is high enough,

$$\text{mean}\,(|d|)_{\downarrow S_x^{(k)}} \approx \text{mean}\,(|d|)_{\downarrow P_x^{(k)}} \approx \frac{s[L] - s[1]}{L - 1}, \tag{1.31}$$

so that equation (1.30) becomes

$$\tilde{Q}^{(k)} = \frac{N_y}{N_x}\frac{P_x^{(k)} - 1}{P_x^{(k)} - 1/N_x}. \tag{1.32}$$

From (1.32) it results that all monotonic signals have almost the same representability characteristic drawn in Figure 1.40 as a line. The circles correspond to experimental data for various monotonic signals like linear, parabolic of various degrees, logarithmic and exponential of various bases, and so forth. Monotonic signals are the best practical case in what concerns the representability characteristic. As results from (1.32) and from the data in Figure 1.40, for large values of the pixel width P_x, the representability characteristic tends asymptotically towards the aspect ratio of the screen:

$$\tilde{Q}^{(k)} \xrightarrow[2^{k-1}\gg 1]{} \frac{N_y}{N_x}. \tag{1.33}$$

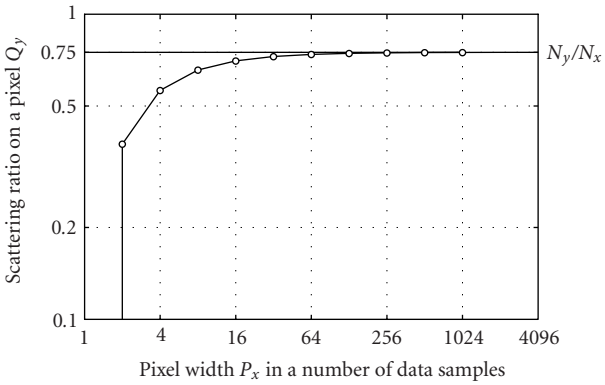


Figure 1.40. Representability diagram (pixel width Px versus average data scattering ratio on a pixel $\tilde{Q}$) for monotonic signals. For the illustration, the length of the signal has been chosen $2^{20} = 1048576$ bp, and the screen size 1024×768 pixels.

1.6.3. Representability practical worst case: uniformly distributed random signals

The theoretical *worst case* from the representability point of view is a hypothetical signal for which the variation between two successive samples is equal to the screen height. A practical worst case is provided by a random signal uniformly distributed on the screen height. The representability characteristic can also be found in closed form for his case. The average variation of the data for the set of $P_x^{(k)}$ samples corresponding to a pixel, that is, the average of the difference between the largest and the smallest values of the samples in the set of $P_x^{(k)}$ random variables uniformly distributed across the screen height expressed in pixels is given by [26]

$$\tilde{Q}^{(k)} = \frac{\tilde{V}_y^{(k)}}{\tilde{P}_y^{(k)}} = N_y \frac{P_x^k - 1}{P_x^k + 1}. \tag{1.34}$$

The representability characteristic is shown in Figure 1.41. The line has been computed analytically using the equation (1.25), while the circles represent data from a Monte Carlo simulation of the uniform distribution of the samples in a range equal to the screen height in data units. For large values of the pixel width, the representability characteristic asymptotically approaches N_y—the vertical size of the screen in pixels:

$$Q^{(k)} \xrightarrow[2^{k-1} \gg 1]{} N_y. \tag{1.35}$$

The monotonic signals and the uniformly distributed random signal provide the practical limiting cases of the framework in which the real-word signal fall.

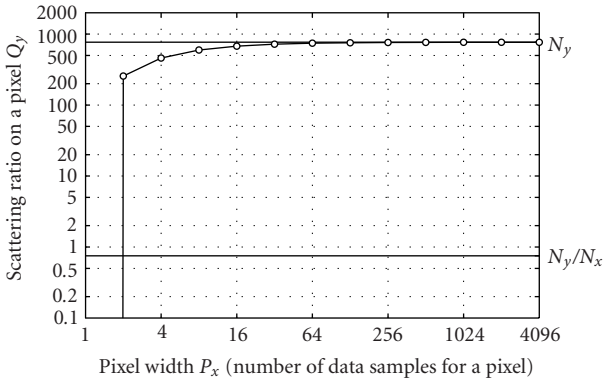


Figure 1.41. Representability diagram for a uniformly distributed random signal (length of the signal $2^{22} = 4194304$ bp, screen size 1024×768 pixels).

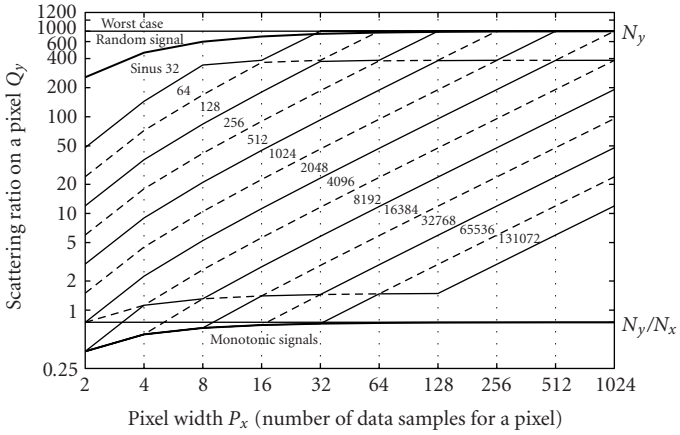


Figure 1.42. Representability diagram for sinus signals of various periods (length of signals 1048576 bp, screen size 1024×768 pixels).

1.6.4. Sine signal representability

To illustrate the behavior of nonmonotonic signals, in Figure 1.42 are given the representability characteristics of several sine functions with periods form 2^5 to 2^{17} samples. As expected, the sine signal behaves as a monotonic signal—the best case—when its period is larger than four times the width of the screen in number of samples, and as the worst case—when the period is lower than the width of the pixels. Two aliasing effects occur in the vicinity of the limiting cases, at levels of the average data scattering ratio equal to twice the best case and half the worst case, respectively. In-between these two levels, the average data scattering ratio varies almost linearly with respect to the pixel width.

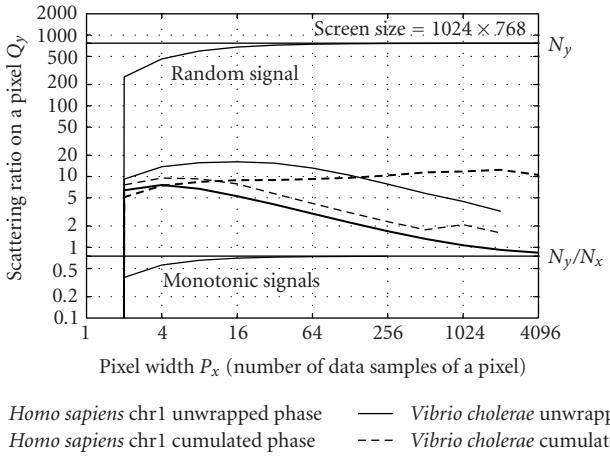


Figure 1.43. Representability diagram $\tilde{Q} = f(P_x)$ for the cumulated and unwrapped phase of the contig NT_004424 [2] of *Homo sapiens* chromosome 1 (length 6,311,978 bp [2, 3, 4]) and of the circular chromosome of *Yersinia pestis* (NC_0003143, length 4,653,728 bp [2, 18]).

1.6.5. Phase signals of genomic signals

Figure 1.43 shows the average data scattering ratio for 6,311,978 base pairs along contig of the first chromosome of *Homo sapiens* (NT 004424 [2]). The results are typical for many other prokaryote and eukaryote genomic signals. The screen size has been considered to be 1024×768 pixels. For the special case of one pixel per sample, for which the variation inside a pixel is zero, the scattering ratio cannot be represented on the logarithmic plot. This case corresponds to an error-free graphic, disregarding the smoothness of the resulting line. For pixels comprising two samples and up to about 16 samples, that is, for DNA segments comprising up to 16384 base pairs, both the cumulated and the unwrapped phase have the average data scattering ratio in the range 5–8, so that the data should be presented taking into account their dispersion. In most cases, this can be done by tracing a couple of lines showing the minimum and maximum values, respectively. When there are only several points apart from the others, the representation can be made by a line corresponding to the average value in a sliding window with the width of a pixel, accompanied by error bars. What is remarkable for the analysis of large scale DNA features is the fact than the average vertical scattering ratio of the signal for a pixel $\tilde{Q}$ becomes less than one, that is, the variation of the signal for the set of samples represented by a pixel becomes less than the pixel height, when the pixel width is larger than about 1450 samples. Obviously, the scale used to represent large scale features of genomic signals is much larger, up to hundreds of thousands of samples per pixel, so that the data can be represented adequately by a single line having the width of only one pixel.

The cumulated phase displays a relatively small variation and, when represented independently, remains with a rather significant dispersion of the samples that requires a presentation similar to the one used for statistical data.

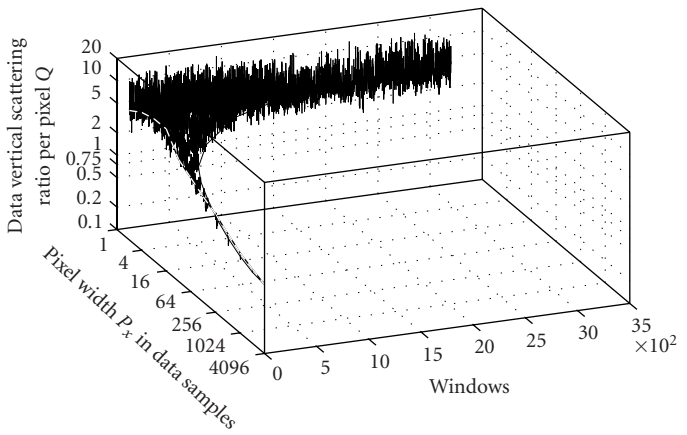


Figure 1.44. 3D representability diagram for the unwrapped phase of the contig NT_004424 [2] of *Homo sapiens* chromosome 1. The average curve in the plane $P_x - Q$ is the representability diagram shown in Figure 1.43.

Figure 1.44 gives a 3D representability diagram of the unwrapped phase of the *Homo sapiens* chromosome 1 contig NT_004424 shown in Figure 1.43. Both the average value of Q—the vertical scattering ratio of the signal on a pixel and the fluctuations of its value in the various windows decrease with the increase of the pixel width P_x.

1.7. Conclusions

This chapter presents results in the analysis of genomic information at the scale of whole chromosomes or whole genomes based on the conversion of genomic sequences into genomic signals, concentrating on the phase analysis.

The most conspicuous result is the linear variation displayed by the unwrapped phase almost along all chromosomes. This feature holds for all the investigated genomes, being shared by both prokaryotes and eukaryotes, while the magnitude and sign of the unwrapped phase slope are specific for each taxon and chromosome. Such a behavior proves a rule similar to Chargaff's rule, but reveals a statistical regularity of the *succession* of the nucleotides—a second-order statistics, not only of the *distribution* of nucleotides—a first order statistics.

This property is related to functions at the scale of whole chromosomes, such as replication, transcription, and crossover. The cumulated phase of the genomic signal of certain prokaryotes also shows a remarkable specific behavior. The comparison between the behavior of the cumulated phase and of the unwrapped phase across the putative origins and termini of the replichores suggests an interesting model for the structure of chromosomes.

The highly regular (linear) shape of the cumulated phase of reoriented ORFs strongly suggests a putative ancestral DNA longitudinal structure from which the current structures have evolved to satisfy restrictions resulting from various chromosome functions.

The analysis of data representability shows that the cumulated phase and the unwrapped phase can be represented adequately as simple graphic lines for very low and large scales, while for medium scales (thousands to tens of thousands of base pairs) statistical descriptions have to be used.

Bibliography

[1] The Genome Data Base, http://gdbwww.gdb.org/, Genome Browser, http://genome.ucsc.edu, European Informatics Institute, http://www.ebl.ac.uk, Ensembl, http://www.ensembl.org.
[2] National Center for Biotechnology Information, National Institutes of Health, National Library of Medicine, http://www.ncbi.nlm.nih.gov/genoms/, ftp://ftp.ncbi.nlm.nih.gov/genoms/, GenBank, http://www.ncbi.nlm.nih.gov/Genbank/index.html.
[3] International Human Genome Sequencing Consortium, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, no. 6822, pp. 860–921, 2001.
[4] J. C. Venter, M. D. Adams, E. W. Myers, et al., "The sequence of the human genome," *Science*, vol. 291, no. 5507, pp. 1304–1351, 2001.
[5] Y. Kawarabayasi, Y. Hino, H. Horikawa, et al., "Complete genome sequence of an aerobic hyperthermophilic crenarchaeon, Aeropyrum pernix K1," *DNA Res.*, vol. 6, no. 2, pp. 83–101, 1999.
[6] RIKEN Genome Exploration Research Group Phase II Team and the FANTOM Consortium, "Functional annotation of a full-length mouse cDNA collection," *Nature*, vol. 409, no. 6821, pp. 685–690, 2001.
[7] Rat Genome Sequencing Consortium, http://www.ncbi.nlm.nih.gov/genoms, 30 August, 2003.
[8] Genome Sequencing Center, Chicken genome, Washington University Medical School, 1 March 2004, http://www.genome.wustl.edu/projects/chicken/.
[9] The C. elegans Sequencing Consortium, "Genome sequence of the nematode C. elegans: a platform for investigating biology," *Science*, vol. 282, no. 5396, pp. 2012–2018, 1998.
[10] A. Theologis, J. R. Ecker, C. J. Palm, et al., "Sequence and analysis of chromosome 1 of the plant Arabidopsis thaliana," *Nature*, vol. 408, no. 6814, pp. 816–820, 2000.
[11] R. A. Alm, L. S. Ling, D. T. Moir, et al., "Genomic-sequence comparison of two unrelated isolates of the human gastric pathogen Helicobacter pylori," *Nature*, vol. 397, no. 6715, pp. 176–180, 1999.
[12] K. Aoki, A. Oguchi, Y. Nagai, et al., Sequence of *Staphylococcus aureus* (strain MW2), direct submission to GenBank," 6 March 2002, National Institute of Technology and Evaluation, Biotechnology Center, Tokyo, Japan, http://www.bio.nite.go.jp/.
[13] T. Baba, F. Takeuchi, M. Kuroda, et al., "Genome and virulence determinants of high virulence community-acquired MRSA," *Lancet*, vol. 359, no. 9320, pp. 1819–1827, 2002.
[14] F. R. Blattner, G. Plunkett III, C. A. Bloch, et al., "The complete genome sequence of Escherichia coli K-12," *Science*, vol. 277, no. 5331, pp. 1453–1474, 1997.
[15] J. Kawai, A. Shinagawa, K. Shibata, et al., "Functional annotation of a full-length mouse cDNA collection," *Nature*, vol. 409, no. 6821, pp. 685–690, 2001.
[16] F. Kunst, N. Ogasawara, I. Moszer, et al., "The complete genome sequence of the gram-positive bacterium Bacillus subtilis," *Nature*, vol. 390, no. 6657, pp. 249–256, 1997.
[17] J. R. Lobry, "Asymmetric substitution patterns in the two DNA strands of bacteria," *Mol. Biol. Evol.*, vol. 13, no. 5, pp. 660–665, 1996.
[18] J. Parkhill, B. W. Wren, N. R. Thomson, et al., "Genome sequence of *Yersinia pestis*, the causative agent of plague," *Nature*, vol. 413, no. 6855, pp. 523–527, 2001.
[19] T. Shimizu, K. Ohtani, H. Hirakawa, et al., "Complete genome sequence of Clostridium perfringens, an anaerobic flesh-eater," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 2, pp. 996–1001, 2002.
[20] J. M. Claverie, "Computational methods for the identification of genes in vertebrate genomic sequences," *Hum. Mol. Genet.*, vol. 6, no. 10, pp. 1735–1744, 1997.
[21] W. F. Doolittle, "Phylogenetic classification and the universal tree," *Science*, vol. 284, no. 5423, pp. 2124–2128, 1999.
[22] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge, UK, 1998.

[23] P. D. Cristea, "Conversion of nucleotides sequences into genomic signals," *J. Cell. Mol. Med.*, vol. 6, no. 2, pp. 279–303, 2002.
[24] P. D. Cristea, "Genetic signal representation and analysis," in *SPIE Conference, International Biomedical Optics Symposium, Molecular Analysis and Informatics (BIOS '02)*, vol. 4623 of *Proceedings of SPIE*, pp. 77–84, San Jose, Calif, USA, January 2002.
[25] P. D. Cristea, "Genomic signals of chromosomes and of concatenated reoriented coding regions," in *SPIE Conference, Biomedical Optics (BIOS '04)*, vol. 5322 of *Proceedings of SPIE*, pp. 29–41, San Jose, Calif, USA, January 2004, Progress in Biomedical Optics and Imaging, Vol. 5, No. 11.
[26] P. D. Cristea, "Representability of genomic signals," in *Proc. 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, San Francisco, Calif, USA, September 2004.
[27] P. D. Cristea, "Genomic signals of reoriented ORFs," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 132–137, 2004, Special issue on genomic signal processing.
[28] P. D. Cristea, "Large scale features in DNA genomic signals," *Signal Process.*, vol. 83, no. 4, pp. 871–888, 2003, Special issue on genomic signal processing.
[29] P. D. Cristea, "Analysis of chromosome genomic signals," in *7th International Symposium on Signal Processing and Its Applications (ISSPA '03)*, pp. 49–52, Paris, France, July 2003.
[30] P. D. Cristea and G. A. Popescu, "Fractal dimension of human chromosome 22," in *The 1st South-East European Symposium on Interdisciplinary Approaches in Fractal Analysis (IAFA '03)*, pp. 131–134, Bucharest, Romania, May 2003.
[31] P. D. Cristea, "Genomic signals for whole chromosomes," in *SPIE Conference, International Biomedical Optics Symposium, Molecular Analysis and Informatics (BIOS '03)*, vol. 4962 of *Proceedings of SPIE*, pp. 194–205, San Jose, Calif, USA, January 2003.
[32] P. D. Cristea, "Large scale features in prokaryote and eukaryote genomic signals," in *9th International Workshop on Systems, Signals and Image Processing (IWSSIP '02)*, Manchester, UK, November 2002.
[33] E. Chargaff, "Structure and function of nucleic acids as cell constituents," *Fed. Proc.*, vol. 10, no. 3, pp. 654–659, 1951.
[34] J. D. Watson and F. H. C. Crick, "A structure for deoxyribose nucleic acid," *Nature*, vol. 171, no. 4356, pp. 737–738, 1953.
[35] D. Anastassiou, "Frequency-domain analysis of biomolecular sequences," *Bioinformatics*, vol. 16, no. 12, pp. 1073–1081, 2000.
[36] D. R. Forsdyke, "Sense in antisense?," *J. Mol. Evol.*, vol. 41, no. 5, pp. 582–586, 1995.
[37] J. M. Freeman, T. N. Plasterer, T. F. Smith, and S. C. Mohr, "Patterns of genome organization in bacteria," *Science*, vol. 279, no. 5358, pp. 1827–1830, 1998.
[38] A. Grigoriev, "Analyzing genomes with cumulative skew diagrams," *Nucleic Acids Res.*, vol. 26, no. 10, pp. 2286–2290, 1998.
[39] J. O. Andersson, W. F. Doolittle, and C. L. Nesbø, "Genomics. Are there bugs in our genome?," *Science*, vol. 292, no. 5523, pp. 1848–1850, 2001.
[40] H. Gee, "A journey into the genome: what's there," *Nature Science Update*, February 2001, http://www.nature.com/news/2001/010215/full/010215-3.html.
[41] M. Kuroda, T. Ohta, I. Uchiyama, et al., "Whole genome sequencing of meticillin-resistant Staphylococcus aureus," *Lancet*, vol. 357, no. 9264, pp. 1225–1240, 2001.

Paul Dan Cristea: Biomedical Engineering Center, University Politehnica of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania

Email: pcristea@ieee.org

2 Gene feature selection

Ioan Tabus and Jaakko Astola

This chapter presents an overview on the classes of methods available for feature selection, paying special attention to the problems typical to microarray data processing, where the number of measured genes (factors) is extremely large, in the order of thousands, and the number of relevant factors is much smaller. The main ingredients needed in the selection of an optimal feature set consist in: the search procedures, the underlying optimality criteria, and the procedures for performance evaluation. We discuss here some of the major classes of procedures which are apparently very different in nature and goals: a typical Bayesian framework, several deterministic settings, and finally information-theoretic methods. Due to space constraints, only the major issues are followed, with the intent to clarify the basic principles and the main options when choosing one of the many existing feature selection methods.

2.1. Introduction

There are two major distinct goals when performing gene feature selection: the first is *discovering the structure* of the genetic network or of the genetic mechanisms responsible for the onset and progress of a disease; the second is eliminating the irrelevant genes from a classification (or prediction) model with the final end of *improving the accuracy* of classification or prediction. While there are many cases when both goals are equally relevant, there are others when only one of them is of primary focus.

This possible distinction of goals is certainly reflected at the methodological level, where the feature selection methods are usually split into two groups: filter methods and wrapper methods [1]. With *the filter methods* [2, 3], the genes are ranked according to some general properties (correlation, mutual information, discriminative power) that are relevant for the prediction or classification problem at hand (e.g., correlation with a disease type), but without making it explicit at this stage what is the particular prediction model that is going to be used subsequently. After ranking of the single genes or of the various groups of genes, a suitable set of genes is identified and proposed as the feature set to be used for all subsequent

analysis tasks. This hints that filter methods should have a more general goal than simply improving a certain type of predictor (e.g., a neural network), but rather should aim at finding the true structure of the interactions recorded by the experimental data, and as such, they would provide a useful set of features for very different classification tools. On the other hand, the *wrapper methods* [1, 4, 5, 6] are more straightforward, since they intend to restrain the set of factors so that the prediction ability of a certain given method is improved. With the wrapper method, the prediction capability of a particular method is investigated for all possible groups of genes (or only for a chosen subset of them, if complete search is computationally too demanding), and the group offering the best performance is declared an optimal set of feature genes, which certainly maximizes the prediction abilities of the studied class of models, but may not be relevant for other classes of models. The same dichotomy between filter and wrapper methods is relevant for the evaluation stage, where the available set of data may be used in various ways for assessing the performance of a given feature set. The above distinction, made in terms of goals and methodology, appears clearly with most of the existing feature selection techniques.

Although ideally feature selection is a main step in the attempt to discover true biological relationships, rarely a feature set is claimed to have full biological relevance without further validation. The apparent optimal or quasioptimal behavior observed over the studied experimental data is only the starting point for more detailed biological experimentation and validation. In light of this, many times, the feature selection procedure ends up proposing *several* likely outstandingly performing feature sets, which can be used later in biological studies [5, 7].

In pattern recognition tasks, the information typically appears as vectors of a large number of variables. We can view the recognition algorithm as operating directly on the variables and consider the variables as features, whence feature selection means selecting a useful subset of variables for further processing. In microarray data analysis, this is often the case, and feature selection essentially means gene selection. Sometimes it is better to draw the line between preprocessing and recognition closer to the end and assume that the actual recognition is done using perhaps quite complicated functions of the original variables. In this situation, feature selection means both selection of the relevant variables and the process of forming suitable informative expressions of groups of variables.

The necessity of feature selection is well documented in the machine learning literature, with examples, where the performance of classification or prediction algorithms degrades quickly if irrelevant features are added, and even if relevant features are added, when they are correlated with the current features. This degradation is even more drastic in the realistic scenarios, where the underlying distribution is not known, and the classification algorithm must estimate the parameters from data; if the number of features is high, the variance of the large number of corresponding parameters is also high, and it becomes attractive to trade off the high number of features (which may be required for a low bias) for a smaller number of parameters, and implicitly a smaller variance (but unfortunately a higher bias). Since the achievable accuracy depends on the size of the available data sets,

a somehow surprising situation occurs: the estimated optimal feature set depends on the size of the training set, being larger when the number of data points is large, but smaller when the available data sets are small.

We start the discussions giving a working definition of the optimal feature set, adapted here from [1], by which we attempt to clarify main issues to be tackled, rather than trying to give a general definition valid across all approaches. The dataset $\mathcal{D} = \{(X_1(t),\dots,X_N(t),Y(t)) \mid t = 1,\dots,n\}$ has n instances of the features and their corresponding class labels $Y \in \{1,\dots,K\}$. The full set of features is formed of all N available features, X_i, $i = 1,\dots,N$, and is denoted by the vector $\mathbf{X} = [X_1,\dots,X_N]^T$, where T denotes transposition. A subset of selected features $\{X_{i_1},\dots,X_{i_k}\}$ will be specified in two equivalent ways: either by the set of selecting indices $A = \{i_1,\dots,i_k\}$, or by a vector $\boldsymbol{\gamma}$ of length N that has zeros everywhere, except the positions $i_1,\dots,i_k$, that is, $\gamma_{i_1} = \cdots = \gamma_{i_k} = 1$, and consequently, the feature set can be denoted as $\mathbf{X}_{\boldsymbol{\gamma}} = [X_{i_1},\dots,X_{i_k}]^T$.

Definition. Given a classifier $g(x_1,\dots,x_r;\boldsymbol{\theta})$ (able to operate on a variable number r of inputs, and depending on the tunable parameter vector $\boldsymbol{\theta}$) and a data set $\mathcal{D}$ with features $X_1,\dots,X_N$ and target Y, sampled from a (unknown) joint distribution, the optimal feature set $\{X_{i_1},\dots,X_{i_k}\}$ is a subset of k features that maximizes the classification accuracy of the classifier $g(X_{i_1},\dots,X_{i_k};\boldsymbol{\theta}^*)$ having the best parameter vector $\boldsymbol{\theta}^*$.

The most sensitive issues in the above definition are the specification of the family of parametric classifiers, the selection of a suitable measure of accuracy, and the estimating accuracy when the underlying distribution is not known.

The organization of the chapter is as follows. In Section 2.2, we present several instances of the feature selection problem under idealistic settings, stressing on the importance of the proper definition of an optimal feature set, dependent on the considered types of models and adopted performance measures. The two important classes of feature selection methods, filter methods and wrapper methods, are discussed in Sections 2.3 and 2.4, respectively. A distinct group of methods is treated in Section 2.5, where the simultaneous search for the best feature set and the best model parameters makes the methods particularly efficient. Finally, in Section 2.6, we discuss the minimum-description-length- (MDL-) based feature selection, which is an information-theoretic method grounded in the fundamental principle of choosing the models according to their description length.

2.2. Feature selection under ideal settings

It will be useful to review various attempts to formally define an optimal feature set under idealized and restricted scenarios. Incorporating all intuitive requirements, which are usually associated to the (common sense) denomination "*feature set,*" will show to be difficult even in the simple case of binary classification problems, indicating that precise qualifications need to be added to make the concept of *optimal feature set* well defined.

2.2.1. Bayes classification

In a typical Bayesian scenario, the set of all features contains N features (real-valued random variables) $\{X_i \mid 1 \le i \le N\}$, which are related to the target value Y (or class label), which is a binary random variable. The full description of this dependency is provided by the joint distribution, specified by the pair (μ, η), where μ is a probability measure for X (i.e., for a set $B \subseteq \mathbb{R}^N$, we have $\mu(B) = P(X \in B)$) and $\eta(x) = P(Y = 1 \mid X = x)$. Using the description (μ, η), we can compute the probability $P((X, Y) \in C)$, where for convenience, we split the set C into the union of two subsets corresponding to $Y = 0$ and $Y = 1$, as follows: $C = (C_0 \times \{0\}) \cup (C_1 \times \{1\})$. Then the joint probability can be evaluated as $P((X, Y) \in C) = \int_{C_0}(1 - \eta(x))\mu(dx) + \int_{C_1} \eta(x)\mu(dx)$, see [8]. In the most ideal case, the joint distribution is known, for the purpose of evaluating the performance of a given classifier. The goal is to find a binary classifier $g(X_{i_1}, \ldots, X_{i_k})$, which is a function of only k of the N features, such that the probability of error $P\{g(X_{i_1}, \ldots, X_{i_k}) \ne Y\}$ is as small as possible. The set of feature genes will be completely determined by the set of indices $A_k = \{i_1, \ldots, i_k\}$. In order not to obscure the following discussion by the particular form of the function g (which may be, e.g., a perceptron, a logistic regression, etc.) we consider here the best possible unrestricted classifier $g^* : \mathbb{R}^k \to \{0, 1\}$, which achieves the infimum of the Bayes' classification error [8]

$$\varepsilon(A_k) = \inf_{g:\mathbb{R}^k \to \{0,1\}} P\{g(X_{i_1}, \ldots, X_{i_k}) \ne Y\}. \tag{2.1}$$

Thus, we can evaluate the performance of a feature set by its Bayesian error. Obviously, $\varepsilon(A_k) \ge \varepsilon(B_{k+1})$ whenever $A_k \subset B_{k+1}$, which expresses the monotonicity property under nesting, a property which is also found with many other performance measures.

In the above setting, the usefulness of a feature set is considered in a plausible way, but the monotonicity under nesting leads to a counterintuitive solution: the full $\mathbf{X}$ is one of the (many possible) best feature sets, because there is no gain in the Bayesian error by restricting the feature set. The nonuniqueness of the best feature set, and the fact that the full feature set $\mathbf{X}$ is already an optimal solution, make the Bayesian error as such a nonattractive measure of optimality. Thus, additional requirements must be added, for example, we may ask which is the smallest size k at which the Bayes error is the best possible, that is, ask for the smallest k^* for which there is a set $A^*_{k^*}$ such that

$$\varepsilon(A^*_{k^*}) = \varepsilon(A_N) = \inf_{g:\mathbb{R}^N \to \{0,1\}} P\{g(X_1, \ldots, X_N) \ne Y\}. \tag{2.2}$$

Unfortunately, for a generic joint distribution of (X, Y), the best set A_{k^*} will be identical to A_N, since, generically, each feature will carry some nonzero information regarding Y. So, the problem "what is the cardinality of the feature set" will receive the trivial answer, "N," for most distributions.

For that reason, further restrictions are needed for getting nontrivial answers, for example, fix the value of k and ask which is the best feature set of cardinality k. This optimality problem is very difficult from a computational viewpoint, since one has to test $\binom{N}{k}$ possible subsets of cardinality k, which is a feasible task only for small values of N and k. Thus, for a practical solution, we should ask if there is a structural property of the optimal Bayes solution which may help in restricting the search. An answer in the negative is offered by the following result, due to Cover [9]: choose an arbitrary ordering of all possible subsets of X, and index them as $B^1, B^2, \ldots, B^{2^N}$; if the consistency constraint $i < j$ for $B^i \subset B^j$ is satisfied (therefore $B^1 = \varnothing$, $B^{2^N} = X$), then there exists a joint distribution of (X, Y) such that

$$\varepsilon(B^1) > \varepsilon(B^2) > \cdots > \varepsilon(B^{2^N}). \tag{2.3}$$

According to this theorem, every possible subset of X with size k can be an optimal feature set A_{k*}^*, and therefore there is no straightforward way to restrict the search for A_{k*}^*. In particular, the following algorithm (dubbed here *Best individual k genes*) in general is doomed to fail: evaluate each feature X_i according to its Bayes error $\varepsilon(\{i\})$ when used as a singleton feature set, and build the candidate set of k features using the best ranking features, according to $\varepsilon(\{i\})$. Even when exhaustive search is computationally unfeasible, heuristical solutions are available and most notably branch-and-bound algorithms based on dynamic programming exist, which exclude from the search some subsets due to the monotonicity under nesting property [10].

The difficulties revealed in defining and finding an optimal feature set for the ideal case of Bayes error using an unrestricted classifier are indicative of the problems that will be faced with the realistic problem, that of finding the feature set for a constrained class of classifiers, and under a finite data set.

2.2.2. Learning classifiers under finite data

We briefly review the problems encountered when looking for a proper feature set by learning under finite data specification, especially under the ideal error-free scenario. To simplify notations, in this section, the features are discrete valued. We explore here various definitions of the intuitive notion of feature *relevance*, and analyze the relevance of features contained in the optimal feature set (the one that has the optimum accuracy).

Let $S_i = \{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_N\}$ be the set of all features except X_i. Let S_i' be a subset of S_i. *Weak relevance* [1] of the feature X_i means that there exist some S_i', x_i, y_i, and s_i' for which $P(X_i = x_i,\ S_i' = s_i') > 0$ such that

$$P(Y = y \mid X_i = x_i,\ S_i' = s_i') \neq P(Y = y \mid S_i' = s_i'), \tag{2.4}$$

Table 2.1

X_1	X_2	X_3	X_4	X_5	Y
0	0	0	1	1	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	1	0	0	0

that is, the probability of a feature given a partial set of features will change if the information regarding feature X_i is withdrawn. Thus, at least for the class label y, the weakly relevant gene contains information which no other gene in S_i' can substitute. The *strong relevance* [1] of a feature means that the feature has weak relevance and in addition, the set S_i' satisfying condition (2.4) is the full feature set, $S_i' = S_i$.

As an example, we consider a space of five binary features, and the case of observing a (target) random variable for which there is an error-free representation as $Y = X_1 \oplus X_2$, and error-free measurements are available from all the feature values, which are connected as follows: $X_4 = \overline{X}_2$ and $X_5 = \overline{X}_3$ [1]. The input space and the corresponding output value are represented in Table 2.1.

We also suppose that all feature vectors compatible with the constraints (there are 8 such vectors) are equiprobable. Thus $P(Y = 0 \mid X_1 = 0,\ X_2 = 0) = P(Y = 0 \mid X_1 = 1,\ X_2 = 1) = P(Y = 1 \mid X_1 = 0,\ X_2 = 1) = P(Y = 1 \mid X_1 = 1,\ X_2 = 0) = 1$. Feature X_1 is strongly relevant, because $P(Y = 0 \mid X_1 = 0,\ X_2 = 0,\ X_3 = 0,\ X_4 = 1,\ X_5 = 1) = 1$, while $P(Y = 0 \mid X_2 = 0,\ X_3 = 0,\ X_4 = 1,\ X_5 = 1) = 1/2$. Further, X_2 is weakly relevant, because $P(Y = 0 \mid X_1 = 0,\ X_2 = 0) = 1$, while $P(Y = 0 \mid X_2 = 0) = 1/2$. Weakly relevant is also X_4, but X_3 and X_5 are irrelevant (i.e., they are neither strongly relevant nor weakly relevant).

At this point, it can be seen that the dependencies of the features affect their relevance with respect to the target, and restraining the feature set can lead to changes of status from weakly relevant to strongly relevant.

An unexpected fact is that the optimal feature set with respect to accuracy may not contain all relevant features. An example of this behavior is in the following scenario: there are two binary features, and all feature combinations are equiprobable. The "true" target is again supposed to be a deterministic function of the features $Y = X_1 \oplus X_2$. Both features are strongly relevant. Suppose now that we want to find an optimal classifier in the family of threshold functions, for example, the classifier needs to be representable as $g(X_1, X_2; \boldsymbol{\theta}) = \delta(\theta_1 X_1 + \theta_2 X_2 > \theta_0)$ for some positive numbers $\theta_0, \theta_1, \theta_2$, where $\delta(\cdot)$ is the Kronecker symbol (equal to 1 if the argument is true, and 0 otherwise). There are two optimal threshold functions: one has $\boldsymbol{\theta} = [0.5 \quad 1 \quad 0]^T$ and it is equal to $g^*(X_1, X_2; \boldsymbol{\theta}) = X_1$; the other has $\boldsymbol{\theta} = [0.5 \quad 0 \quad 1]^T$ and it is equal to $g^*(X_1, X_2; \boldsymbol{\theta}) = X_2$, and both

have accuracy 2/4 (each makes two mistakes), and all other threshold functions have lower accuracy. It is thus possible that under a restricted class of classifiers some strongly relevant features are left out of the feature set that has optimal accuracy. Despite this, in practice, it is necessary to restrict the class of classifiers, but the above phenomenon shows that it is important to do the restriction in a way that does not prevent the useful information entering the classification. The reasons why one may wish to restrict the class of allowed classifiers are at least twofold: with a restricted class, the search space can be considerably reduced, partly easing the computational burden; or even better, under restricted classes, closed-form optimal solutions may exist, completely eliminating the need for search.

2.2.3. Learning classifiers in realistic scenarios

In the really important practical scenario, a finite experimental data set, $\mathcal{D} = \{(X_1(t), \ldots, X_N(t), Y(t)) \mid 1 \le t \le n\}$, is given and a restricted family of classifiers is specified. One is interested in finding the optimal feature set, which minimizes the estimated accuracy of the classifier over the data set. Since now the underlying probability distribution of the random variable (X, Y) is not known, the accuracy of the classifier has to be estimated from the data themselves, and the estimate of an optimal feature set will depend on the chosen estimation method. In particular, the estimated accuracy will be subject to the common variance-bias tradeoff: on one extreme, the estimate will be variant but unbiased, on the other less variant but biased. Not only the found optimal feature set for a given classifier is affected by the estimation procedure, but also the attempt to compare the accuracy reached by various classifiers becomes difficult. A common recipe for lowering the variance of the estimate is to use the available data in a circular fashion in turns as training and test set; however this may lead to another, more subtle form of overfitting, making the estimated errors overly optimistic.

There is a large body of literature focusing on the application of various classification methods for the analysis of microarray data, including discussion of the feature selection step, many of which are briefly reviewed in the next sections. In general, any classification method can be used, and will present specific issues, in conjunction with a feature selection method. The support vector machines were successfully used in [4, 11, 12, 13, 14, 15, 16, 17]. Various tree-classification methods have been used and compared for microarray data in [2, 18]. The k-nearest-neighbor classification algorithms have been used in [19, 20], and the Fisher linear discrimination has been tested in [2, 21]. Many other methods are briefly reviewed in the rest of the chapter.

2.3. Filter methods

Several intuitive methods are used to assess the dependency between the features and the targets, such as mutual information, or correlation coefficient. One of the most often used measures for finding the discriminative power of a gene j is the

ratio

$$\frac{\mathrm{BSS}(j)}{\mathrm{WSS}(j)} = \frac{\sum_{t=1}^{n}\sum_{k=1}^{K}\delta\big(Y(t)=k\big)\big(\overline{X}_j^{(k)}-\overline{X}_j\big)^2}{\sum_{t=1}^{n}\sum_{k=1}^{K}\delta\big(Y(t)=k\big)\big(X_j(t)-\overline{X}_j^{(k)}\big)^2}, \tag{2.5}$$

where $\delta(\cdot)$ is the Kronecker symbol; $\overline{X} = (1/n)\sum_t X_j(t)$ is the sample mean of feature j, $\overline{X}_j^{(k)} = (1/n^{(k)})\sum_{t|Y(t)=k} X_j(t)$ is the sample conditional mean of the feature j given class k, $(\sigma_j^{(k)})^2 = (1/n^{(k)})\sum_{t|Y(t)=k}(X_j(t)-\overline{X}_j^{(k)})^2$ is the sample variance of feature j conditional to class k, and $n^{(k)}$ is the number of data points from the training set $\mathcal{D}$ falling in class k. The criterion (2.5) was used as a filter method to select the feature space for an experimental comparison of a large number of different classifiers in microarray gene expression data [2].

Simple manipulations show that

$$\frac{\mathrm{BSS}(j)}{\mathrm{WSS}(j)} = \frac{\sum_k n_k\big(\overline{X}_j^{(k)}-\overline{X}_j\big)^2}{\sum_k n_k\big(\sigma_j^{(k)}\big)^2}, \tag{2.6}$$

which explains the intuition behind the discriminative power of this measure; the total sum of squares $\mathrm{TSS}(j) = \sum_t (X_j(t)-\overline{X}_j)^2$ can be decomposed into two terms, $\mathrm{TSS}(j) = \mathrm{BSS}(j) + \mathrm{WSS}(j)$, where the first, $\mathrm{BSS}(j)$, shows the spread of the class averages with respect to the joint average (and ideally, we want this spread to be as large as possible) while the second, $\mathrm{WSS}(j)$, shows the spread of the points inside each class (and ideally, this should be as low as possible).

For the case of only two classes ($K = 2$) and equal number of instances in each class ($n^{(1)} = n^{(2)}$), the measure (2.5) can be seen to be proportional to the Fisher discriminant ratio

$$\mathrm{FDR}(j) = \frac{\big(\overline{X}_j^{(1)}-\overline{X}_j^{(2)}\big)^2}{\big(\sigma_j^{(1)}\big)^2+\big(\sigma_j^{(2)}\big)^2}. \tag{2.7}$$

The main disadvantage of filter methods is that they look at each feature independently when considering its merit in discriminating the target. If one feature X_{i_1} will be deemed to be highly discriminative for the target, so will be another feature X_j if X_{i_1} and X_j are highly correlated. Thus, when the feature set is redundant, a filter method will recommend the inclusion of all features which are individually highly discriminative, even though the information brought in by a redundant feature is not improving at all the prediction accuracy of the classifier. Including redundant features may degrade the accuracy of a predictor, and to avoid that, a preprocessing stage of clustering for identifying the highly correlated genes may be needed. Other preprocessing, like principal component analysis (PCA), which is intensively used in signal processing for tasks as signal compression or signal analysis, may also reduce, very efficiently, the number of features to be considered, if the features are highly correlated.

More elaborated filter techniques were tested in tandem with wrapper methods on the microarray data, like the Markov-Blanket filter [3].

In a more loose sense, all methods for finding feature sets can be used as filters for other methods, for example, for restraining the feature sets before resorting to methods that are more computationally intensive.

2.4. Wrapper methods

If the final goal is to improve the classification accuracy, the method for feature selection should consider the particular method for classification, with its biases and tradeoffs regarding the evaluation of performance. For this reason, the most natural procedure is to cascade the feature selection process with the classifier design, and to iterate both in a loop until the best performance is achieved, the feature selection process being comparable with a wrapper over the classifier design and evaluation.

2.4.1. A generic wrapper method

The classifier design enters the wrapper algorithm as a black box, so the wrapper approach can be applied to a wide family of classification algorithms. The algorithm for a generic wrapper is presented in Figure 2.1, while one of the blocks from Figure 2.1, namely the block *Find best feature set and parameters*, is detailed in Figure 2.2.

As presented, any wrapper method must consist of two nested cross-validation loops. In the most outer loop, the available data is split into training and test data (in Figure 2.1, the split is in four equal parts). There are many ways to make the split, and for each of them, the algorithm in Figure 2.1 is run once, resulting in an estimate of the performance offered by the block *Find best feature set and parameters*, with a given classification method. The overall performance will be evaluated by the average of all errors $\hat{\varepsilon}$ over a large number of different fourfold splits of the available data.

The blocks *Find best feature set and parameters* are the inner cross-validation loops, and one such block is illustrated in Figure 2.2 for a threefold split of the training data (remark that the available data here is $\mathcal{D}_A, \mathcal{D}_B, \mathcal{D}_C$, that is, only the training data from the outer cross-validation loop). The selected feature set and the classifier parameters should perform well, not only with the data used for design, they have also to generalize well over unseen data. To achieve this, at the stage of search of the feature set, trust should be given only to the performance over a set which was not seen when optimizing the parameters of the classifier. Thus, the search for a feature set is directed by a cross-validation error, as illustrated in Figure 2.2 with a threefold error evaluation.

An important overfitting situation was signaled repeatedly from the early attempts of finding feature sets (see the discussions in [1, 4]): the cross-validation error that was used for guiding the search process for the best feature should not be used as final performance measure of the chosen feature set for comparisons

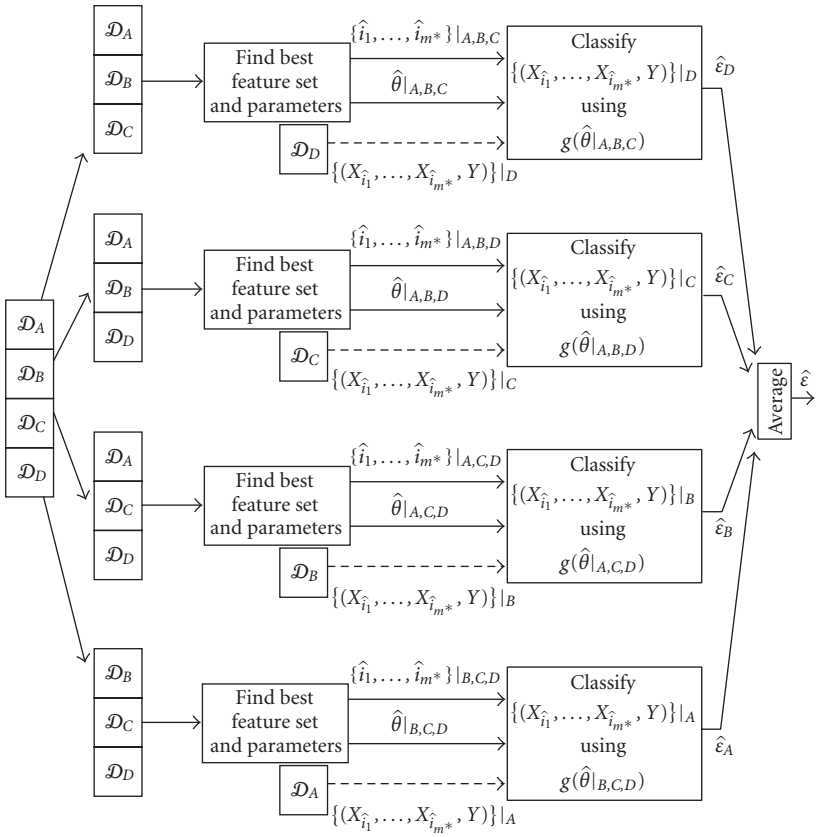


Figure 2.1. Overall estimation of the classification accuracy within a fourfold cross-validation experiment. The blocks "Find the best feature set and parameters" operate on three of the folds (as depicted in Figure 2.2) while the classification accuracy is estimated over the fourth fold, to avoid overfitting.

with other methods, since it is overly optimistic, in other words it is not fair to use the full data set $\mathcal{D}$ for selecting the optimal features $X_{\hat{i}_1}, \ldots, X_{\hat{i}_{k^*}}$, and then use the same data for the computation of the cross-validation error. Thus, the error $\hat{\varepsilon}$ represented in Figure 2.1 should be used for reporting performance results.

2.4.2. The search for best feature set

The search process can be performed according to several heuristics, since in general exhaustive search is not computationally feasible. The greedy methods are usually based either on growing the optimal feature set starting from the empty set (in the *forward selection* methods), or on iteratively removing features starting from the full feature set (in the *backward* methods). The simplest search engine working in a forward selection mode is the *hill climbing*, where new candidate sets

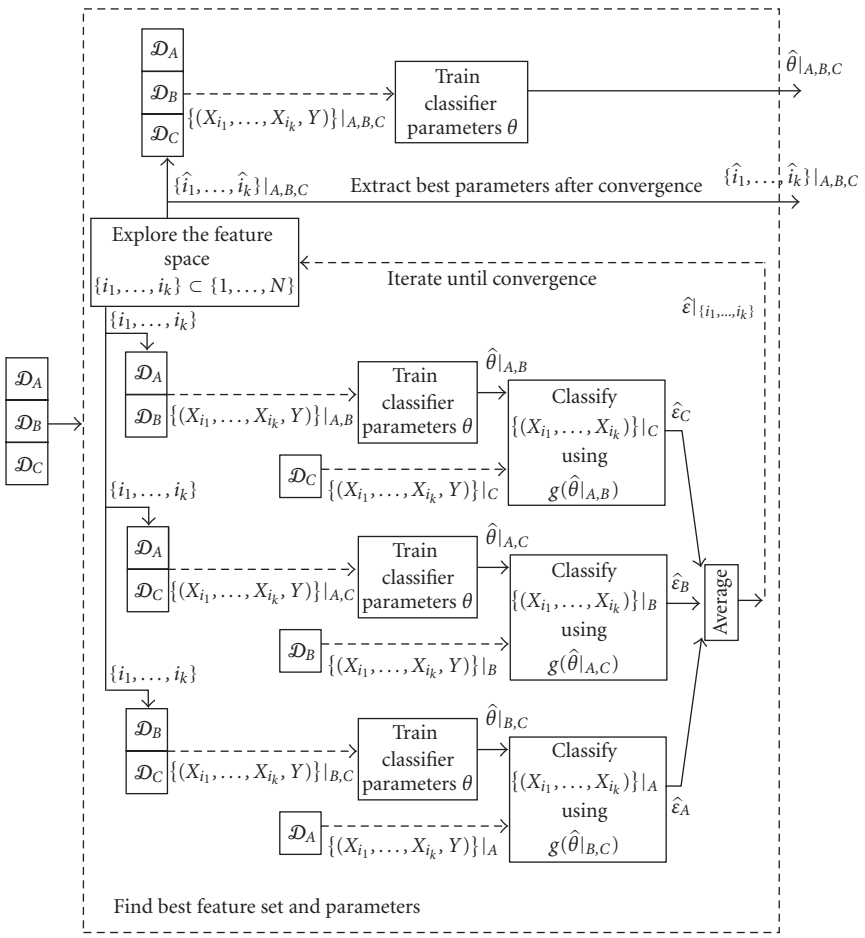


Figure 2.2. The estimation of the classification accuracy of a given feature set $X_{i_1},\ldots,X_{i_m}$ by a threefold cross-validation experiment: the training set $\mathcal{D}=\{(X_1,\ldots,X_N,Y)\}$ is split into three equally sized subsets $\mathcal{D}_A,\mathcal{D}_B,\mathcal{D}_C$. The classifier parameters are fitted to the data provided by two of the subsets of $\mathcal{D}$, and the accuracy is estimated over the remaining subset, and the process is repeated for all ways of taking two out of three subsets. The computed average classification error $\hat{\varepsilon}|_{\{i_1,\ldots,i_m\}}$ is used to direct the search for the most accurate feature set $X_{i_1^*},\ldots,X_{i_{m^*}^*}$ and its size m^*.

are obtained by combining the current feature set with each of the remaining features, and the performance is evaluated for each of the new candidates, the best performing being selected as new current feature set and the process continues until none of the candidates has performance better than the current set. A more laborious forward selection process is *the best first search* engine, where at each iteration, all previously tested sets are checked for the best performer which was not already expanded, the best performer is then combined with the remaining features, and the process continues as long as needed (a fair stopping criterion is

to check whether in the last k steps no improvement was obtained). In return for the increased complexity, the best first search algorithm sometimes reaches better accuracies than hill climbing, but there are also reports of no improvement situations. The mentioned forward selection searches have the disadvantage known as nesting property, namely that a feature, once entered into a feature set, it can not be excluded later. To eliminate the nesting drawback, in the sequential forward floating selection (SFFS) after each inclusion of a new feature a backtracking process is started to check for possible improvements by removal of an existing feature, and when all paying-off removals have been executed, the algorithm resumes to adding a new feature. With this new search policy, one checks more subsets, but it takes a longer time to run. It was widely believed that this computational expense is compensated by better results. However, a recent paper, [6], pointed out that the comparisons in some publications reporting better results were obtained using the wrong cross-validation methodology; when performed in the correct cross-validation framework, the results of the simple forward selection search are as good or better than SFFS in many cases. The results of [6] are of even more general interest, they show that selecting the wrong cross-validation evaluation environment is not benign, it does not lead only to too optimistic reported errors, uniformly for all methods, but it may also lead to (involuntarily) dishonest inversions of ranking of the tested methods.

2.4.3. Optimization criteria and regularization

The search for the best set of genes $\{X_{i_1},\dots,X_{i_k}\}$ and for the parameter vector $\hat{\boldsymbol{\theta}}$ of the best classifier $g(X_{i_1},\dots,X_{i_k};\hat{\boldsymbol{\theta}})$ should be directed by an as relevant criterion as possible. The sample variance

$$J(\boldsymbol{\theta},\boldsymbol{\gamma}) = \frac{1}{n}\sum_{t=1}^{n}\left(Y(t) - g\left(X_{i_1}(t),\dots,X_{i_k}(t);\boldsymbol{\theta}\right)\right)^2 \tag{2.8}$$

is relevant in itself and has the obvious merit of simplicity and ease of evaluation, though usually requires additional constraints for making the solution unique.

The particular case of the perceptron, which can be defined as a linear combiner followed by a threshold, that is, $g(X_{i_1},\dots,X_{i_m};\boldsymbol{\theta}) = T(\sum_{j=1}^{m} X_{i_j}\theta_j)$ with $T(x) = 0$ for $x < 0$, and $T(x) = 1$ for $x \geq 0$, was successfully considered in several gene features methods [5, 7]. In order to obtain a closed form solution of (2.8), the classifier is further approximated by the simpler linear combiner (or linear regression) during the optimization stage, as follows: denoting the vector of observations $\mathbf{X}_{\gamma}(t) = [X_{i_1}(t),\dots,X_{i_k}(t)]^T$, the least squares solution minimizing (2.8) for the linear model $g(\mathbf{X}_{\boldsymbol{\gamma}}(t),\boldsymbol{\theta}) = \boldsymbol{\theta}^T\mathbf{X}_{\boldsymbol{\gamma}}(t)$ is

$$\hat{\boldsymbol{\theta}} = \left(\sum_{t=1}^{n}\mathbf{X}_{\boldsymbol{\gamma}}(t)\mathbf{X}_{\boldsymbol{\gamma}}(t)^T\right)^{-1}\sum_{t=1}^{n}\mathbf{X}_{\boldsymbol{\gamma}}(t)Y(t) \tag{2.9}$$

and it is unique if $\hat{R} = \sum_{t=1}^{n} \mathbf{X}_\gamma(t)\mathbf{X}_\gamma(t)^T$ is nonsingular (otherwise, $\hat{R}^{-1}$ should be taken as a notation for the pseudoinverse of $\hat{R}$). Unfortunately, for large values of k (and in particular for all $k > n$) which are of interest in gene feature selection problems, the matrix $\hat{R}$ is singular, signaling that many optimal vectors $\boldsymbol{\theta}$ exist that reach the lowest possible value of the criterion (2.8), $J(\hat{\boldsymbol{\theta}}, \boldsymbol{\gamma}) = 0$. Avoiding these degenerate situations can be achieved by standard methods, known as regularization techniques, the most used being ridge regression and cross-validation.

2.4.3.1. Ridge regression

With ridge regression, a penalty term $\sigma^2\|\boldsymbol{\theta}\|^2$ is added to $J(\boldsymbol{\theta})$ to prevent solutions with too large parameters, and σ^2 is a weighting on how strong this penalty should be. The minimum value of the new criterion

$$J(\boldsymbol{\theta}, \boldsymbol{\gamma}) + \sigma^2\|\boldsymbol{\theta}\|^2 = \frac{1}{n}\sum_{t=1}^{n}\left(Y(t) - \boldsymbol{\theta}^T\mathbf{X}_{\boldsymbol{\gamma}}(t)\right)^2 + \sigma^2\|\boldsymbol{\theta}\|^2 \tag{2.10}$$

is reached by

$$\hat{\boldsymbol{\theta}} = \left(n\sigma^2 I + \sum_{t=1}^{n}\mathbf{X}_{\boldsymbol{\gamma}}(t)\mathbf{X}_{\boldsymbol{\gamma}}(t)^T\right)^{-1}\sum_{t=1}^{n}\mathbf{X}_{\boldsymbol{\gamma}}(t)Y(t), \tag{2.11}$$

which also reminds us of a well-known regularization method for solving ill-conditioned system of equations.

The same solution (2.11) is an optimal solution for a very interesting related problem: consider new feature X_i', obtained by adding white noise with variance σ^2 to each feature, that is, $X_i'(t) = X_i(t) + e_i(t)$, where $e_i(t)$ is white and independent of $X_j(t)$ and $Y(t)$, and consider the problem of minimizing the expected value

$$EJ(\boldsymbol{\theta}, \boldsymbol{\gamma}) = E\sum_{t=1}^{n}\left(Y(t) - \boldsymbol{\theta}^T\mathbf{X}'_{\boldsymbol{\gamma}}(t)\right)^2, \tag{2.12}$$

where E denotes the expectation operator. The optimal solution minimizing (2.12) is long known to be exactly (2.11) (see, e.g., [22, 23]). The attractive interpretation of the newly generated features X_i' is that of an additional training set (of infinite size), which transforms the set of points of coordinates $X_{\boldsymbol{\gamma}}$ into a set of spheres centered at $X_{\boldsymbol{\gamma}}$, the common radius of the spheres being controlled by σ^2. With original training sets of small size, too many "perfect" solutions exist, if there is a perfect solution, then there is an infinity of hyperplanes separating perfectly the points, and they may be considered artifacts due to the degeneracy of the criterion at the current sample size. The additional training set helps in removing those degenerate solutions. This principle is applied and further developed in [7], where learning of the parameter vector $\boldsymbol{\theta}$ of the perceptron classifier is approximated by learning of the parameter vector $\boldsymbol{\theta}$ of the associated linear classifier, in order to accelerate the training process.

2.4.3.2. Cross-validation as a regularization technique

Cross-validation is intuitively a proper method for evaluating the performance of a classifier in more objective way than the sample variance of the classification errors over the training set [24, 25, 26]. But apart of that, cross-validation is also a tool for regularizing the solution of the criterion (2.8). In the case of linear regression, the cross-validation criterion can be computed in a closed form, making it attractive, for example, for evaluating the performance of the closed-form solutions offered by the ridge regression methods, for various values of the parameter σ^2.

2.4.3.3. Coefficient of determination

The coefficient of determination (COD) is a normalized version of the classification error variance, in which the performance of the tested classifier is normalized with respect to the performance of a classifier that uses no features [27], that is,

$$\mathrm{COD} = \frac{\sigma_{\varnothing}^2 - \sigma_{\boldsymbol{y}}^2(\boldsymbol{\theta}_{\boldsymbol{y}})}{\sigma_{\varnothing}^2}, \tag{2.13}$$

where $\sigma_{\varnothing}^2$, in the case of linear classifiers, is the variance of the class label Y (i.e., the cost of predicting the class by the average over the full dataset). Different types of COD can be defined for each different type of variance: a true COD corresponds to true variances, estimated in a Bayesian setting, where the true joint distribution is known; a resubstitution COD corresponds to the variance of the resubstitution errors, obtained over the full data set for the best predictor $\boldsymbol{\theta}_{\boldsymbol{y}}^*$ designed over the full data set; and a cross-validation COD corresponding to the cross-validation errors. The COD is more advantageous than the variance itself when discussing the performance of a predictor, since it is normalized to have the maximum value of 1. This normalization does not affect otherwise the ranking of performance of two predictors as compared to the ranking performed by the variance itself, but proves useful in presenting comparatively the results for different data sets [5, 27, 28].

2.4.3.4. Information-theoretic criteria

Apart from the criteria based on the variance of the errors, many information-theoretic methods define different criteria for the amount of information about the target contained in a set of genes. Any of these criteria can be used when performing the search for the best set of genes in the context of Figure 2.1. However, a true wrapper method will need to use the inner cross-validation loops depicted in Figure 2.2 for avoiding the overfitting of the model, while the information-theoretic criteria [29] such as MDL or Akaike information criterion (AIC), or Bayes information criterion (BIC) are in themselves protected against overfitting, and therefore it is no need to iterate over the expensive cross-validation loops presented in Figure 2.2.

Information-theoretic measures are used in different forms by various procedures for feature selections and predictor design [30, 31, 32, 33, 34, 35, 36, 37, 38]. In Section 2.6, we review the MDL principle and its use for feature selection.

2.4.4. Methodology for practical evaluation

The evaluation of feature extraction procedure is often confused with the evaluation of a classification procedure, and for that reason, it is carried on with the wrong methodology.

In all gene selection experiments, a data set $\mathcal{D}$ is available, containing measurements of the N gene expressions $\{X_1(t), \ldots, X_N(t) \mid 1 \le t \le n\}$ for each of the n patients and the class labels $\{Y_1(t), \ldots, Y_N(t) \mid 1 \le t \le n\}$ for each patient. Two important problems can be distinguished.

(1) *Predictor evaluation.* A set of selected gene features $X_{\boldsymbol{\gamma}_0} = \{X_{i_1}, \ldots, X_{i_k}\}$ has been specified a priori, through the selection vector $\boldsymbol{\gamma}_0$, (but it was not inferred from a gene selection from the data $\mathcal{D}$!) and the goal is to compare different predictors classes $g(X_{\boldsymbol{\gamma}_0}, \boldsymbol{\theta})$, for example, compare perceptrons against multilayer perceptrons. A methodology which avoids overfitting is the one presented in Figure 2.2, and the quantity $\widehat{\varepsilon}|_{\boldsymbol{\gamma}_0}$ is a good measure of accuracy achieved by different predictors classes (the number of folds in cross-validation can certainly be chosen other than three, but kept the same if several predictors are compared).

(2) *Evaluation of a procedure for feature selection.* If a certain procedure for feature selection needs to be compared with another one, the cross-validation scenario should be the one presented in Figure 2.1 (with a number of folds conveniently chosen, four was used only for illustration) and the quantity $\widehat{\varepsilon}$ is a good measure of accuracy. With the leave-one-out cross-validation, the data is folded in n, such that each data point is left out during the training and used as a test point. If the number of errors in a leave-one-out cross-validation experiment is zero for all tested feature selection methods, one needs to reduce the number of folds in order to test the generalization power of different schemes in a more stringent experiment. By reducing the number of folds, the number of possible splits of the data into training and test also increases, thus a larger number of tests can be carried on.

2.5. Simultaneously searching the best feature set and best model parameters

2.5.1. Bayesian approaches

In the Bayesian setting, a model of the conditional probability $P(Y_t = 1 \mid \boldsymbol{\theta}, \boldsymbol{\gamma})$ is specified and some prior distributions are suitably chosen, such that the posterior probability $P(\boldsymbol{\theta}, \boldsymbol{\gamma} \mid \mathcal{D})$ can be either computed by integration, or can be sampled, and then the optimal feature set is taken to be the maximum a posteriori estimation $\boldsymbol{\gamma}^*$ that reaches the maximum of $P(\boldsymbol{\theta}, \boldsymbol{\gamma} \mid \mathcal{D})$.

There are several possible models to be considered for the conditional probability $P(Y_t = 1 \mid \boldsymbol{\theta}, \boldsymbol{\gamma})$: logistic regression [39, 40] and probit regression [41, 42, 43, 44].

The probit regression model of the conditional probability is $P(Y_t = 1 \mid \boldsymbol{\theta}, \boldsymbol{\gamma}) = \Phi(\mathbf{X}_{\boldsymbol{\gamma}}^T \boldsymbol{\theta})$, where the so-called probit link function $\Phi(\cdot)$ is the normal cumulative distribution $\Phi(z) = (1/\sqrt{2\pi}) \int_{-\infty}^{z} \exp(-x^2/2) dx$. Due to the computationally demanding integrations required, the use of this model was quite limited until the introduction of Gibbs sampling algorithms based on data augmentation (for a recent account, see [45]). In addition to $\boldsymbol{\theta}, \boldsymbol{\gamma}$, a set of latent (unobserved) variables $Z(1), \ldots, Z(n)$ is introduced, with $Z(t) \sim N(\mathbf{X}_{\boldsymbol{\gamma}}^T(t)\boldsymbol{\theta}, 1)$, that is, the link to the regressors is provided by $Z(t) = \mathbf{X}_{\boldsymbol{\gamma}}^T(t)\boldsymbol{\theta} + e_i$, where $e_i \sim N(0,1)$. Prior distributions are selected such that they are easily interpretable and the resulting Markov chain Monte Carlo (MCMC) simulations have a good computational speed and reliable model fitting. The unknowns (Z, θ, γ) will be obtained by Gibbs sampling from the posterior distribution.

A prior distributions $\boldsymbol{\gamma}$ should reflect the need to select small sets of features, and a simple way to enforce a small number of units in $\boldsymbol{\gamma}$ is to take a small value for $P(\gamma_i = 1) = \pi_i$, and consider that the elements of $\boldsymbol{\gamma}$ are independent one of another, so $P(\boldsymbol{\gamma}) = \prod_{i=1}^{N} \pi_i^{1-\gamma_i} (1 - \pi_i)^{\gamma_i}$. Given $\boldsymbol{\gamma}$, the prior on the regressor vector is $\boldsymbol{\theta}_{\boldsymbol{\gamma}} \sim N(0, c(X_{\boldsymbol{\gamma}}^T X_{\boldsymbol{\gamma}})^{-1})$, where $X_{\boldsymbol{\gamma}}$ is the matrix having as column t the vector $\mathbf{X}_{\boldsymbol{\gamma}}(t)$ and c is a constant to be chosen by the user.

By applying Bayesian linear model theory, the following conditional distributions can be computed after evaluating several integrals: $P(\boldsymbol{\gamma} \mid Z)$, $P(Z \mid \boldsymbol{\theta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma})$, and $P(\boldsymbol{\theta} \mid Z, \boldsymbol{\gamma})$ [42]. By iteratively drawing $\boldsymbol{\gamma}, Z$, and $\boldsymbol{\theta}$ from these distributions, one obtains an MCMC simulation, which will allow to make inference even in this case, in which there is no explicit expression for the posterior distribution. After an initial burn-in period, it is assumed that the samples are taken from the posterior distribution, and one can compute the relative number of times each gene was selected, for making a decision about including it in the feature set [42]. An extension to the multiclass classification problems can be obtained by generalizing the probit model to multinomial probit models [44]. After finding the average frequency $\overline{\gamma}_i$ by which each gene was selected during the MCMC simulation, the most important genes for a (sub-)optimal feature set are chosen, for example, those for which $\overline{\gamma}_i$ exceeds a certain threshold. With the selected genes, either the probit model is used to make classifications or other models may be used, if they give better accuracy [44].

A different optimization approach is the expectation-maximization (EM) method in which Laplacian priors for γ and θ are chosen to reflect our need of a sparse solution. The hyperparameters τ, ρ of these distributions, and the latent variables $Z(1), \ldots, Z(n)$ are used in an EM optimization algorithm, where in the first step (E-step), the expected value $Q(\boldsymbol{\theta}, \boldsymbol{\gamma} \mid \widehat{\boldsymbol{\theta}}^{\ell}, \hat{\boldsymbol{\gamma}}^{\ell})$ of the posterior $\log P(\boldsymbol{\theta}, \boldsymbol{\gamma} \mid D, Z, \tau, \rho)$ is computed conditioned on the current estimates $\hat{\boldsymbol{\gamma}}^{\ell}$ and $\widehat{\boldsymbol{\theta}}^{\ell}$, and in the second step (M-step), new estimates are obtained by the maximization $\widehat{\boldsymbol{\theta}}^{\ell+1}, \hat{\boldsymbol{\gamma}}^{\ell+1} = \arg\max_{\boldsymbol{\theta}, \boldsymbol{\gamma}} Q(\boldsymbol{\theta}, \boldsymbol{\gamma} \mid \widehat{\boldsymbol{\theta}}^{\ell}, \hat{\boldsymbol{\gamma}}^{\ell})$ [41].

2.5.2. Deterministic criteria allowing one-step solutions

We consider here the binary classification problem and the task is to find a perceptron which optimally separates the classes. If the problem has a perfect solution $\boldsymbol{\theta}^*$, the hyperplane $z = \boldsymbol{\theta}^{*T}\mathbf{X}+b$ will be such that the available data points $(\mathbf{X}(t), Y(t))$ satisfy $Y(t) = \text{sign}(z(t))$ and $Y(t)(\boldsymbol{\theta}^{*T}\mathbf{X}(t)+b) \geq 1$. To encourage a sparse solution $\boldsymbol{\theta}^*$, that is, most of the elements of $\boldsymbol{\theta}^*$ to be zero, the following criterion should be minimized: $\|\boldsymbol{\theta}\|_0 \triangleq \sum_i \theta_i^0 = \sum_i \gamma_i$, subject to the constraint that all points are classified correctly (with the convention $0^0 = 0$).

$$\begin{aligned} &\min_{\boldsymbol{\theta},b} \|\boldsymbol{\theta}\|_0 \\ &\text{s.t.} \quad Y(t)\left(\boldsymbol{\theta}^T\mathbf{X}(t) + b\right) \geq 1, \quad 1 \leq t \leq n. \end{aligned} \tag{2.14}$$

The problem has to be relaxed in several ways, to make it tractable and to account for more realistic, noisy, situations.

The foremost modification is the substitution of the ℓ^0 norm with the ℓ^1 norm in the criterion, to transform the problem into a convex optimization problem, for which efficient numerical solutions exist. As shown in [46], for a large number of problems, the transformation of an ℓ^0 optimization problem into an ℓ^1 optimization problem leads to a solution for the latter, identical to the solution of the former, provided that the solution is sufficiently sparse. Even though in our general scenario, the sufficient conditions established in [46] may not hold, and so the solutions of the two problems may be different, the resulting solution of the ℓ^1 problem has a high degree of sparsity. The situation is completely different with the ℓ^2 norm, where a lot of small (nonzero) values for the θ_i's are encouraged by their downweighting in the criterion through taking the square.

A second major modification accounts for possible classification errors, and in order to minimize their effect, the nonnegative slack variables $\xi(t)$ are introduced to relax each constraint, $Y(t)(\boldsymbol{\theta}^T\mathbf{X}(t)+b) \geq 1-\xi(t)$, and the penalty term $\sum_{t=1}^n \xi(t)$ is added to the criterion, weighted by a constant C [47]:

$$\begin{aligned} &\min_{\boldsymbol{\theta},b} \|\boldsymbol{\theta}\|_1 + C\sum_{t=1}^{n} \xi(t) \\ &\text{s.t.} \quad Y(t)\left(\boldsymbol{\theta}^T\mathbf{X}(t) + b\right) \geq 1 - \xi(t), \\ &\qquad \xi(t) \geq 0, \quad 1 \leq t \leq n. \end{aligned} \tag{2.15}$$

The similarity between (2.10) and (2.15) is obvious when $n\sigma^2 = 1/C$: one considers the sum of squared errors regularized by the ℓ^2 norm of the parameter vector, the other considers the sum of absolute values of the errors regularized by the ℓ^1 norm of the parameter vector.

The differences come from the fact that by directly optimizing (2.10), the solution is in general nonsparse, so one has to optimize for the huge number of 2^N combinatorial choices of $\boldsymbol{\gamma}$ in order to find the best sparse solution, while solving (2.15) can be realized in a single optimization step, as described next. To allow both positive and negative elements in θ, the positive variables u_i and

v_i are introduced such that $\theta_i = u_i - v_i$ and $|\theta_i| = u_i + v_i$ and thus (2.15) is transformed into the new optimization problem [47]

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{v},b} \sum_{i=1}^{N} (u_i + v_i) + C \sum_{t=1}^{n} \xi(t) \quad \text{s.t.} \quad Y(t)\big((\mathbf{u}-\mathbf{v})^T \mathbf{X}(t) + b\big) \geq 1 - \xi(t), \\ &\xi(t) \geq 0, \quad 1 \leq t \leq n, \\ &u_i \geq 0, \quad v_i \geq 0, \quad 1 \leq i \leq N, \end{aligned} \tag{2.16}$$

which is a standard linear programming (LP) problem with inequality constraints.

Therefore, once the regularization constant C is chosen, solving the optimization problem (2.16) for the sparse vector of parameters $\boldsymbol{\theta}$ (and hence for the optimum $\boldsymbol{\gamma}$, which has the same pattern of zeros as $\boldsymbol{\theta}$) can be done in a very fast way, using standard numerical optimization tools.

2.5.3. Joint feature clustering and classification

The feature selection problem refers to extracting an informative and nonredundant set from the existing features. The somehow related problem of forming new features (e.g., by linear combination of some existing genes) is traditionally considered a distinct problem and it was well studied in the past. However, there are many methods having as a preliminary stage (before proceeding to classification) the formation of new features, notable examples being principle component analysis and clustering. The *SimClust* algorithm described in [32], is one such approach and is mentioned here because it is well related to the method described in Section 2.6. *SimClust* solves a combined problem: simultaneously find clusters (groups of related genes) and classify the samples using as classification features the "average" genes which are the centers of the obtained clusters. To this goal, an MDL cost is associated to describing the clusters, and another MDL cost is associated to the regression model for optimal scoring. Relative weights are set such that the two MDL costs have the same importance when the sample size is increased. We remark that the method is an involved mixture of techniques, for example, prior to starting the computational demanding part algorithm, a filter approach is used to restrain the set of genes to only T genes, those having the largest values of between-to-within-class sum of squares (2.5).

2.6. Minimum-description-length-based feature selection

MDL was used as a basis for statistical inference in a large number of problems dealing with finding the structure of models. Its bases were laid down about 25 years ago in [48] inspired by the work on complexity in [49, 50, 51], and further refined in a number of papers [52, 53, 54]. As a fundamental principle, it states that given a model class, the best model should be chosen based on its ability to represent the data in the most compact form, that is, with the MDL. Evaluating the description code length is closely related to probabilistic modeling, since if one

knows the probability $P(x^n)$ of a string x^n, the optimal description length can be shown to be $-\log_2 P(x^n)$. For making inference, the value itself of the description length is sufficient, but it is worth noting that this value can really be achieved in practice when using arithmetic coding (within a very good precision), so the description length of sequences or parameters in this section really refers to short descriptions of the data or parameters, descriptions that can be decoded into the original sequences. The overall MDL is based on efficient lossless codes, and we can show at anytime a real message encoding the data with the claimed description length, since MDL takes care of all the costs involved by a real description. In contrast, many other methods of feature selection based on entropy or mutual information systematically neglect parts of the costs of encoding the data, that is, when the empirical entropy of a sequence is used for inference, the cost of encoding the probability model is not included, and thus comparisons across models of different complexities may be unfair.

2.6.1. MDL using two-part codes

In order to apply MDL to feature selection, first a model class should be chosen and then the total description length needs to be computed. If a two-part code is used, as in [48], the total description length can be defined as the sum of the description of the model parameters and the description of the data given the model parameters. This instance of MDL was applied to genomic data for the first time in [33] in the following way. Given a class of predictors, for example, perceptrons, the optimal predictor of the target $\{Y(t) \mid 1 \le t \le n\}$ given the input data $\{\mathbf{X}(t) \mid 1 \le t \le n\}$ is first found and its optimal parameters are denoted $\boldsymbol{\theta}_{\boldsymbol{\gamma}}^*$. The prediction errors $\{\varepsilon(t) = Y(t) - T(\mathbf{X}^T(t)\boldsymbol{\theta}_{\boldsymbol{\gamma}}^*) \mid 1 \le t \le n\}$ obtained with the optimal predictor $\boldsymbol{\theta}_{\boldsymbol{\gamma}}^*$ are then encoded by a simple code, for example, by encoding the locations of the nonzero errors, using L_ε bits. For encoding the optimum perceptron parameters $\boldsymbol{\theta}_{\boldsymbol{\gamma}}^*$, the most simple code will be constructed by assuming that all distinct perceptrons having $k = \sum_i \gamma_i$ inputs (there are $n_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}}$ of them) are equally likely, and it will require $L_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}} = \log_2(n_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}})$ bits. The total description length of the two-part code, $L_{\text{tot}}(\boldsymbol{\gamma}) = L_\varepsilon + L_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}}$, can be used as a criterion for discriminating between different structures $\boldsymbol{\gamma}$. The penalty introduced by encoding the parameters of the model, $L_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}}$, is clearly increasing with the perceptron size, k, and therefore, it constitutes a penalty on the complexity of the model. If the order k of a model is too large, the optimal predictions obtained with the model will be very good, thus the cost of encoding the errors, L_ε, may be very small or even zero, but such a model is discouraged by the complexity penalty. On the other hand, a too simple model will be unable to predict the target well, given the input, so the cost of the errors will be high, even though the cost of the model (its complexity $L_{\boldsymbol{\theta}_{\boldsymbol{\gamma}}}$) will be small. So the proper balance between modeling power and model complexity is established by the MDL principle, to guide the selection of the right structure $\boldsymbol{\gamma}$. The nice theoretical properties of MDL for selecting the structure of models have been established for a wide class of models, and the simplicity of its use makes it an attractive tool for feature selection.

2.6.2. MDL using normalized maximum likelihood models

Although the pioneering work on MDL with two-part codes proved to be practical and well justified theoretically, the later developments in universal data compression have introduced more efficient codes to be used for the evaluation of description lengths [29, 54]. In the rest of this section, we present the approach introduced in [34], to which we refer for more details and proofs.

The target, or class label, is a string $Y^n = Y(1),\dots,Y(n)$ of n realizations of the random variable Y, taking values in the set $\{0,\dots,M-1\}$. Each $Y(t)$ is observed together with the k-tuple (a column vector) $\mathbf{X}(t) = [X_{i_1}(t)\cdots X_{i_k}(t)]^T$, where the features are discrete valued $X_i(t) \in \{0,\dots,n_q-1\}$, being quantized to n_q levels. The sequence of regressors $\mathbf{X}^n = \mathbf{X}(1),\dots,\mathbf{X}(n)$ contains vectors which may occur repeatedly, and let K denote the number of distinct vectors, $K \le n$, which means that the various vectors $\mathbf{X}(t)$ belong to a finite set denoted $\{\mathbf{b}_1,\dots,\mathbf{b}_K\}$. In the end, we need to collect counts of the symbols conditioned on a specific value of the regression vector in order to estimate model parameters, so we introduce the following notations: let $\mathcal{I}_j$ denote the set of indices at which $\mathbf{b}_j$ is found in the sequence $\mathbf{X}(1),\dots,\mathbf{X}(n)$; that is, $\mathcal{I}_j = \{t : \mathbf{X}(t) = \mathbf{b}_j\}$, which has cardinality $M_j = |\mathcal{I}_j|$. The count $m_q^j = |\{i : y_i = q,\ i \in \mathcal{I}_j\}|$ is the number of observations $Y(t) = q \in \{0,\dots,M-1\}$ at which the regressor vector was $\mathbf{X}(t) = \mathbf{b}_j$, and the counts should obey $m_0^j + \cdots + m_{M-1}^j = M_j$.

The probability distributions $P(Y^n \mid \mathbf{X}^n; \boldsymbol{\eta}, \boldsymbol{\nu})$ of the class labels Y^n conditioned on a given value of the regressor $\mathbf{X}^n$ are parameterized by the sets of parameters $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$. We call the model a discrete regression model, since y takes values from a discrete set, and the joint observations $\mathbf{X}$ take values also in a discrete set.

The number K of different vectors $\mathbf{b}_1,\dots,\mathbf{b}_K$ appearing in the regressor sequence $\mathbf{X}(1),\dots,\mathbf{X}(n)$ can be large, and if it is close to n, it will be difficult to use at each of the regressors $\mathbf{b}_i$ a conditional multinomial model with distinct parameters, since not enough observations will be available to estimate the probabilities of the symbols $Y(t)$ conditional on $\mathbf{X}(t) = \mathbf{b}_i$. Because of this dilution phenomenon, pooling of the frequencies of occurrence of $Y(t)$ at different context together will be beneficial, but pooling should be performed after reordering the counts at a given $\mathbf{b}_i$ in a decreasing sequence.

The permutations $\nu_i(\cdot) \in \Upsilon_M$ (where Υ_M denotes the set of $M!$ permutations of the set $\mathcal{Y} = \{0,1,\dots,M-1\}$) can be used to reorder the class labels $j \in \mathcal{Y}$ such that the frequencies of occurrence of the class labels (observed at the time moments t with $\mathbf{X}(t) = \mathbf{b}$) are arranged in decreasing order.

To make clear the reordering, we introduce a new string Z^n, obtained from the class labels Y^n by use of the set of permutations $\boldsymbol{\nu} = (\nu_1(\cdot),\dots,\nu_K(\cdot))$ as follows:

$$\begin{aligned} Z(t) &= \nu_\ell^{-1}(Y(t)), \\ Y(t) &= \nu_\ell(Z(t)), \end{aligned} \tag{2.17}$$

where ℓ is the index for which $\mathbf{X}(t) = \mathbf{b}_\ell$ and $\nu_\ell(\cdot)$, $\nu_\ell^{-1}(\cdot)$ are a permutation, and its inverse, respectively. Since $\nu_\ell(\cdot)$ is a permutation of $0,\dots,M-1$,

the transformation is reversible, that is, one can recover $Y(t)$ from $Z(t)$. The realigned string $\{Z^n\}$ is further modeled as a multinomial trial process with parameters $P(Z=0)=\eta_0,\ldots,P(Z=M-1)=\eta_{M-1}$. The symbol i is observed in the string Z^n exactly $\sum_{\ell=1}^{K} m^{\ell}_{v_\ell(i)}$ times, and thus the probability of the class label string is given by

$$\begin{aligned} P(Y^n \mid \mathbf{X}^n;\boldsymbol{\eta},\boldsymbol{v}) &= P(Z^n(\boldsymbol{v});\boldsymbol{\eta},\boldsymbol{v}) \\ &= \eta_0^{\sum_{\ell=1}^{K} m^{\ell}_{v_\ell(0)}} \cdots \eta_{M-1}^{\sum_{\ell=1}^{K} m^{\ell}_{v_\ell(M-1)}}, \end{aligned} \tag{2.18}$$

where the sequence of new data Z^n is determined by the set of permutations $\boldsymbol{v}=\{v_i(\cdot): i=1,\ldots,K\}$ as parameters, and the multinomial parameters of the sequence $Z^n(\boldsymbol{v})$ are grouped in the vector $\boldsymbol{\eta}=(\eta_0,\ldots,\eta_{M-1})$.

To make the set of pairs $(\boldsymbol{\eta},\boldsymbol{v})$ nonredundant, the vector $\boldsymbol{\eta}=(\eta_0,\ldots,\eta_{M-1})$ needs to be restricted such that $\eta_0\geq\eta_1\geq\cdots\geq\eta_{M-1}$ and it can be shown that with this constraint, there is no reduction in the flexibility of the model. Finally, based on the above consideration, the model class named here *discrete regression* is formalized as

$$\begin{aligned} \mathcal{M}(\boldsymbol{\eta},k,\boldsymbol{v}) = \Bigg\{ & P(Y^n \mid \mathbf{X}^n;\boldsymbol{\eta},\boldsymbol{v}) : \boldsymbol{\eta}\in[0,1]^M;\ \eta_0\geq\eta_1\geq\cdots\geq\eta_{M-1}; \\ & \sum_{i=0}^{M-1}\eta_i = 1;\ \boldsymbol{v}\in(\Upsilon_M)^K \Bigg\}, \end{aligned} \tag{2.19}$$

where k is the dimensionality of the regression vectors $\mathbf{X}(t)$ and K is the number of distinct vectors in the sequence $\mathbf{X}^n$. The key parameter to be determined during the feature selection process is the number of features k. In the following, the optimal codes for the specified class of models will be obtained, and by computing the optimal description length for various values of k, the MDL principle will be used to determine the optimal k^*.

The optimal description length for strings y^n, using the class $\mathcal{M}(\boldsymbol{\eta},k,\boldsymbol{v})$, can be computed by constructing first the normalized maximum likelihood (NML) model $q(y^n \mid \mathbf{x}^n)$ for the class, and then taking the description length to be $\mathcal{L}(y^n \mid \mathbf{x}^n) = -\log_2 q(y^n \mid \mathbf{x}^n)$. The NML model for a class of probability models can be obtained by first computing the maximized likelihood $P(y^n \mid \mathbf{x}^n;\hat{\boldsymbol{\eta}}(y^n),\hat{\boldsymbol{v}}(y^n))$ using the ML parameters $\hat{\boldsymbol{\eta}}(y^n),\hat{\boldsymbol{v}}(y^n)$, and then normalizing it to a probability distribution $q(y^n \mid \mathbf{x}^n)$. The optimality properties of the NML model for universal data compression have been established in [54].

To compute the ML parameters, the maximization problem is split into two subproblems, first optimize $\boldsymbol{v}$ for a given $\boldsymbol{\eta}$, then optimize $\boldsymbol{\eta}$ for the optimum $\boldsymbol{v}^*(\boldsymbol{\eta})$:

$$\max_{\boldsymbol{\eta}}\Big[\max_{\boldsymbol{v}} P(y^n \mid \mathbf{x}^n;\boldsymbol{\eta},\boldsymbol{v})\Big]. \tag{2.20}$$

The first stage

$$\max_{\boldsymbol{\nu}} P(y^n \mid \mathbf{x}^n; \boldsymbol{\eta}, \boldsymbol{\nu}) = \max_{\nu_1(\cdot)\cdots\nu_K(\cdot)} \prod_{\ell=1}^{K} \eta_0^{m^{\ell}_{\nu_\ell(0)}} \cdots \eta_{M-1}^{m^{\ell}_{\nu_\ell(M-1)}} \tag{2.21}$$

can be immediately seen to decouple into K independent subproblems, one for each permutation $\nu_\ell(\cdot)$, and the optimal permutation is the permutation $\hat{\nu}_\ell(\cdot)$ for which $m^{\ell}_{\hat{\nu}_\ell(0)} \geq m^{\ell}_{\hat{\nu}_\ell(1)} \geq \cdots m^{\ell}_{\hat{\nu}_\ell(M-1)}$. The permutations $\hat{\nu}_\ell(\cdot)$ are the ML set of permutations $\hat{\boldsymbol{\nu}}$, no matter what the values of $\boldsymbol{\eta}$ are [34].

By ordering decreasingly the sequence of numbers $m^j_0, \ldots, m^j_{M-1}$, a new sequence can be defined: $\hat{n}^j_0 = m^j_{(M-1)}, \ldots, \hat{n}^j_{M-1} = m^j_{(0)}$, where the standard notation for the order statistics is used. The number of occurrences of the dominant symbol in each of the sets $\{y_i : i \in \mathcal{I}_1\}, \ldots, \{y_i : i \in \mathcal{I}_K\}$ is collected and the final pooled counts are obtained as $n^*_0 = \hat{n}^1_0 + \cdots + \hat{n}^K_0$. Similarly, denote by $n^*_j = \hat{n}^1_j + \cdots + \hat{n}^K_j$ the total number of occurrences of the jth dominant symbol in each of the sets $\{y_i : i \in \mathcal{I}_1\}, \ldots, \{y_i : i \in \mathcal{I}_K\}$.

By performing the outer maximization in (2.20), the optimal parameters $\hat{\boldsymbol{\eta}}$ turn out to be $\hat{\eta}_i = n^*_i/n$, which is consistent with the assumed ranking $\eta_0 \geq \eta_1 \geq \cdots \eta_{M-1}$. The counts $n^*_i(Y^n)$ depend on Y^n in a complicated manner through the order statistics of the counts $m^{\ell}_i(Y^n)$.

Since the ML values of the parameters are now available, the NML model in the model class $\mathcal{M}(\boldsymbol{\eta}, k, \boldsymbol{\nu})$ can be defined as follows:

$$\begin{aligned} q(y^n \mid \mathbf{x}^n) &= \frac{P(y^n \mid \mathbf{x}^n; \hat{\boldsymbol{\eta}}(y^n), \hat{\boldsymbol{\nu}}(y^n))}{C_n(M_1, \ldots, M_K)}, \\ C_n(M_1, \ldots, M_K) &= \sum_{w^n \in \{0,\ldots,M-1\}^n} P(w^n \mid \mathbf{x}^n; \hat{\boldsymbol{\eta}}(w^n), \hat{\boldsymbol{\nu}}(w^n)), \\ &= \sum_{w^n \in \{0,\ldots,M-1\}^n} \prod_{i=0}^{M-1} \left(\frac{n^*_i(w^n)}{n}\right)^{n^*_i(w^n)}. \end{aligned} \tag{2.22}$$

The computation of the normalizing constant directly from (2.22) is almost impossible for most gene expression data, since it requires the summation of M^n terms, but a more practical approach can be found, as described next.

The computations needed in (2.22) can be rearranged as a single sum as follows:

$$\begin{aligned} &C_n(M_1, \ldots, M_K) \\ &\quad = \sum_{n^*_0 + n^*_1 + \cdots + n^*_{M-1} = n} S_{M_1,\ldots,M_K}(n^*_0, n^*_1, \ldots, n^*_{M-1}) \prod_{\ell=0}^{M-1} \left(\frac{n^*_\ell}{n}\right)^{n^*_\ell}, \end{aligned} \tag{2.23}$$

where $S_{M_1,\ldots,M_K}(n^*_0, n^*_1, \ldots, n^*_{M-1})$ denotes the number of strings w^n having the same $n^*_0(w^n), \ldots, n^*_M(w^n)$. The summation needs to be done over the set of numbers obeying $n^*_0 \geq n^*_1 \geq \cdots \geq n^*_{M-1}$, which can be enforced by defining $S_{M_1,\ldots,M_K}(n^*_0,$

$n_1^*, \ldots, n_{M-1}^*) = 0$ for all the strings $n_0^*, n_1^*, \ldots, n_{M-1}^*$, which are not decreasing strings. The numbers $S_{M_1,\ldots,M_K}(n_0^*, n_1^*, \ldots, n_{M-1}^*)$ can be computed recursively in K, according to the recurrence formula

$$S_{M_1,\ldots,M_K}(n_0^*, n_1^*, \ldots, n_{M-1}^*) = \sum_{i_0+i_1+\cdots+i_{M-1}=M_K} S_{M_1,\ldots,M_{K-1}}(n_0^* - i_0, n_1^* - i_1, \ldots, n_{M-1}^* - i_{M-1}) h(i_0, \ldots, i_{M-1}), \tag{2.24}$$

where $h(i_0, \ldots, i_{M-1})$ denotes the number of ways in which a string having $n' = M_1 + \cdots + M_{K-1}$ letters and $K-1$ distinct regressor vectors can be extended to a string having $n = M_1 + \cdots + M_K$ letters and K distinct regressor vectors, such that in the set $\{y_i : i \in \mathcal{I}_K\}$, the counts of symbols are $i_0, \ldots, i_{M-1}$, regardless of order; it is also required that $i_0 \geq i_1 \geq \cdots \geq i_{M-1}$. The newly introduced convolving sequences $h()$ are defined as

$$h(i_0, \ldots, i_{M-1}) = \binom{i_0 + \cdots + i_{M-1}}{i_0, \ldots, i_{M-1}} \binom{k_0 + \cdots + k_{r-1}}{k_0, \ldots, k_{r-1}} \tag{2.25}$$

in the case of a decreasing sequence of arguments $i_1 \geq \cdots \geq i_{M-1}$, while for all other arguments ($i_0, \ldots, i_{M-1}$ not being decreasing) the sequence is $h(i_0, \ldots, i_{M-1}) = 0$. In (2.25), r is the number of distinct values in the string $i_0, \ldots, i_{M-1}$ and $k_0, \ldots, k_{r-1}$ is the number of repetitions of the distinct values in the string $i_0, \ldots, i_{M-1}$, respectively. For example, with the arguments $i_0 = 6$, $i_1 = 4$, $i_2 = 4$, $i_3 = 3$, $i_4 = 3$, the values occurring in (2.25) are $r = 3$, $k_0 = 2$, $k_1 = 2$, $k_2 = 1$.

The computation of $\mathcal{L}(y^n \mid \mathbf{x}^n)$ can be accomplished very fast, the computation of the normalization constant is most demanding, but its evaluation with (2.23), by means of the convolution sums (2.24), is very fast, its implementation in Matlab is run in less than 1 second.

The search for the best feature set, the one that minimizes the description length $\mathcal{L}(y^n \mid \mathbf{x}^n)$, can be performed in any of the traditional ways for wrapper methods, as described in Section 2.4.2.

The classification model $\hat{y} = g(x_{i_1}, \ldots, x_{i_k})$, discovered during the feature selection process, can be extended to cases which were absent in the training set, to obtain a well-defined classifier. For unseen cases, the decision is taken according to the class labels (votes) of the nearest neighbors at Hamming distance d, where d is the smallest value at which a class label is a definite winner, see, for example, [36].

The presented MDL method can be considered to be a wrapper method where the optimization criterion is not the classification accuracy of a specific classifier, but the description length achieved when using the information from the feature set. The selection process is motivated by an information-theoretic principle, and thus the method can be seen as a powerful tool for discovering informative feature sets, which are very likely to have biological significance. However, the method can be used also as a filter stage, after which the best classifier in a certain class can be easily designed and tested for classification accuracy.

2.7. Conclusions

Feature selection is an involved process, which needs knowledge of the available techniques for guiding which tool to be used and for assessing correctly the performance for the application at hand. An impressive number of studies of feature selection techniques for gene expression data has shown that relevant biological information can be gathered using various feature selection techniques, at a computational cost which is affordable with the current computer technology. Future studies will most likely reduce even further the computational cost of the methods, making it possible to compare larger candidate feature sets. As another challenge for the future, the biological interpretation of the feature sets needs to be integrated within the feature selection methods themselves, and not used as they are now, just as a validation stage after the feature selection process was finished.

Bibliography

[1] R. Kohavi and G. John, "Wrapper for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[2] S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," Tech. Rep. 576, Department of Statistics, University of California, Berkely, Calif, USA, 2000.

[3] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in *Proc. 18th International Conference on Machine Learning*, Morgan Kaufmann, San Mateo, Calif, USA, June 2001.

[4] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[5] S. Kim, E. R. Dougherty, Y. Chen, et al., "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.

[6] J. Reunanen, "Overfitting in making comparisons between variable selection methods," *J. Mach. Learn. Res.*, vol. 3, Special Issue on Variable and Feature Selection, pp. 1371–1382, 2003.

[7] S. Kim, E. R. Dougherty, J. Barrera, Y. Chen, M. L. Bittner, and J. M. Trent, "Strong feature sets from small samples," *J. Comput. Biol.*, vol. 9, no. 1, pp. 127–146, 2002.

[8] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31 of *Applications of Mathematics (New York)*, Springer-Verlag, New York, 1996.

[9] T. M. Cover and J. M. van Campenhout, "On the possible orderings in the measurement selection problem," *IEEE Trans. Systems Man Cybernet.*, vol. SMC-7, no. 9, pp. 657–661, 1977.

[10] P. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Computers*, vol. 26, no. 9, pp. 917–922, 1977.

[11] J. Jaeger, R. Sengupta, and W. L. Ruzzo, "Improved gene selection for classification of microarrays," in *Proc. Pacific Symposium on Biocomputing*, pp. 53–64, Kauai, Hawaii, January 2003.

[12] P. Pavlidis, J. Weston, J. Cai, and W. S. Noble, "Learning gene functional classifications from multiple data types," *J. Comput. Biol.*, vol. 9, no. 2, pp. 401–411, 2002.

[13] A. Rakotomamonjy, "Variable selection using SVM-based criteria," *J. Mach. Learn. Res.*, vol. 3, no. 7-8, pp. 1357–1370, 2003, Special issue on variable and feature selection.

[14] B. Schölkopf, I. Guyon, and J. Weston, "Statistical learning and kernel methods in bioinformatics," in *NATO Advanced Studies Institute on Artificial Intelligence and Heuristics Methods for Bioinformatics*, San Miniato, Italy, October 2001.

[15] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," in *Advances in Neural Information Processing Systems 13*, pp. 668–674, MIT Press, Cambridge, Mass, USA, 2000.

[16] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *J. Mach. Learn. Res.*, vol. 3, Special Issue on Variable and Feature Selection, pp. 1439–1461, 2003.

[17] X. Zhang and W. H. Wong, "Recursive sample classification and gene selection based on SVM: method and software description," Tech. Rep., Department of Biostatistics, Harvard School of Public Health, Cambridge, Mass, USA, 2001.
[18] I. Inza, B. Sierra, R. Blanco, and P. Larrañaga, "Gene selection by sequential search wrapper approaches in microarray cancer class prediction," *Journal of Intelligent and Fuzzy Systems*, vol. 12, no. 1, pp. 25–34, 2002.
[19] G. Fuller, K. Hess, C. Mircean, et al., "Human glioma diagnosis from gene expression data," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., pp. 241–256, Kluwer Academic Publishers, Boston, Mass, USA, 2002.
[20] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, vol. 17, no. 12, pp. 1131–1142, 2001.
[21] M. Xiong, W. Li, J. Zhao, L. Jin, and E. Boerwinkle, "Feature (gene) selection in gene expression-based tumor classification," *Mol. Genet. Metab.*, vol. 73, no. 3, pp. 239–247, 2001.
[22] S. Haykin, *Adaptive Filter Theory*, Prentice Hall International, Englewood Cliffs, NJ, USA, 2nd edition, 1991.
[23] T. Söderström and P. Stoica, *System Identification*, Prentice Hall, New York, NY, USA, 1989.
[24] S. Geisser, "The predictive sample reuse method with applications," *J. Amer. Statist. Assoc.*, vol. 70, no. 350, pp. 320–328, 1975.
[25] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Statist. Soc. Ser. B*, vol. 36, no. 1, pp. 111–147, 1974.
[26] M. Stone, "Asymptotics for and against cross-validation," *Biometrika*, vol. 64, no. 1, pp. 29–35, 1977.
[27] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Process.*, vol. 80, no. 10, pp. 2219–2235, 2000.
[28] R. F. Hashimoto, E. R. Dougherty, M. Brun, Z. Z. Zhou, M. L. Bittner, and J. M. Trent, "Efficient selection of feature sets possessing high coefficients of determination based on incremental determinations," *Signal Process.*, vol. 83, no. 4, pp. 695–712, 2003, Special issue on genomic signal processing.
[29] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2743–2760, 1998, Special commemorative issue: information theory 1948–1998.
[30] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," in *Proc. IEEE Computational Systems Bioinformatics Conference (CSB '03)*, pp. 523–528, Stanford, Calif, USA, August 2003.
[31] C. Furlanello, M. Serafini, S. Merler, and G. Jurman, "Entropy-based gene ranking without selection bias for the predictive classification of microarray data," *BMC Bioinformatics*, vol. 4, no. 1, pp. 1–54, 2003.
[32] R. Jornsten and B. Yu, "Simultaneous gene clustering and subset selection for sample classification via MDL," *Bioinformatics*, vol. 19, no. 9, pp. 1100–1109, 2003.
[33] I. Tabus and J. Astola, "On the use of MDL principle in gene expression prediction," *EURASIP J. Appl. Signal Process.*, vol. 2001, no. 4, pp. 297–303, 2001.
[34] I. Tabus, J. Rissanen, and J. Astola, "Classification and feature gene selection using the normalized maximum likelihood model for discrete regression," *Signal Process.*, vol. 83, no. 4, pp. 713–727, 2003, Special issue on genomic signal processing.
[35] I. Tabus, C. Mircean, W. Zhang, I. Shmulevich, and J. Astola, "Transcriptome-based glioma classification using informative gene set," in *Genomic and Molecular Neuro-Oncology*, W. Zhang and G. Fuller, Eds., pp. 205–220, Jones and Bartlett Publishers, Boston, Mass, USA, 2003.
[36] I. Tabus, J. Rissanen, and J. Astola, "Normalized maximum likelihood models for Boolean regression with application to prediction and classification in genomics," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., pp. 173–196, Kluwer Academic Publishers, Boston, Mass, USA, 2002.
[37] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *J. Mach. Learn. Res.*, vol. 3, no. 7-8, pp. 1415–1438, 2003, Special issue on variable and feature selection.

[38] X. Zhou, X. Wang, and E. R. Dougherty, "Construction of genomic networks using mutual-information clustering and reversible-jump Markov-chain-Monte-Carlo predictor design," *Signal Process.*, vol. 83, no. 4, pp. 745–761, 2003, Special issue on genomic signal processing.
[39] B. Krishnapuram, A. J. Hartemink, and L. Carin, "Applying logistic regression and RVM to achieve accurate probabilistic cancer diagnosis from gene expression profiles," in *Workshop on Genomic Signal Processing and Statistics (GENSIPS '02)*, Raleigh, NC, USA, October 2002.
[40] W. Li and I. Grosse, "Gene selection criterion for discriminant microarray data analysis based on extreme value distributions," in *International Conference on Research in Computational Molecular Biology (RECOMB '03)*, pp. 217–223, Berlin, Germany, April 2003.
[41] B. Krishnapuram, L. Carin, and A. J. Hartemink, "Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data," *J. Comput. Biol.*, vol. 11, no. 2-3, pp. 227–242, 2004.
[42] K. E. Lee, N. Sha, E. R. Dougherty, M. Vannucci, and B. K. Mallick, "Gene selection: a Bayesian variable selection approach," *Bioinformatics*, vol. 19, no. 1, pp. 90–97, 2003.
[43] N. Sha, M. Vannucci, P. J. Brown, M. K. Trower, G. Amphlett, and F. Falciani, "Gene selection in arthritis classification with large-scale microarray expression profiles," *Comparative and Functional Genomics*, vol. 4, no. 2, pp. 171–181, 2003.
[44] X. Zhou, X. Wang, and E. R. Dougherty, "Gene prediction using multinomial probit regression with Bayesian gene selection," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 115–124, 2004.
[45] K. Imai and D. A. van Dyk, "A Bayesian analysis of the multinomial probit model using marginal data augmentation," *J. Econometrics*, vol. 124, no. 2, pp. 311–334, 2005.
[46] D. L. Donoho and M. Elad, "Optimally-sparse representation in general (non-orthogonal) dictionaries via ℓ^1 minimization," Tech. Rep., Stanford University, Stanford, Calif, USA, 2002.
[47] C. Bhattacharyya, L. R. Grate, A. Rizki, et al., "Simultaneous relevant feature identification and classification in high-dimensional spaces: Application to molecular profiling data," *Signal Process.*, vol. 83, no. 4, pp. 729–743, 2003, Special issue on genomic signal processing.
[48] J. Rissanen, "Modelling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.
[49] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Probl. Inf. Transm.*, vol. 1, no. 1, pp. 1–7, 1965.
[50] R. J. Solomonoff, "A formal theory of inductive inference. I," *Information and Control*, vol. 7, pp. 1–22, 1964.
[51] R. J. Solomonoff, "A formal theory of inductive inference. II," *Information and Control*, vol. 7, pp. 224–254, 1964.
[52] J. Rissanen, "Stochastic complexity and modeling," *Ann. Statist.*, vol. 14, no. 3, pp. 1080–1100, 1986.
[53] J. Rissanen, "Fisher information and stochastic complexity," *IEEE Trans. Inform. Theory*, vol. 42, no. 1, pp. 40–47, 1996.
[54] J. Rissanen, "Strong optimality of the normalized ML models as universal codes and information in data," *IEEE Trans. Inform. Theory*, vol. 47, no. 5, pp. 1712–1717, 2001.

Ioan Tabus: Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland

Email: tabus@cs.tu.fi

Jaakko Astola: Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland

Email: jta@cs.tu.fi

3 Classification

Ulisses Braga-Neto and Edward R. Dougherty

3.1. Introduction

Classification plays an important role in genomic signal analysis. For instance, cDNA microarrays can provide expression measurements for thousands of genes at once, and a key goal is to perform classification via different expression patterns. This requires designing a classifier (decision function) that takes a vector of gene expression levels as input, and outputs a class label that predicts the class containing the input vector. Classification can be between different kinds of cancer, different stages of tumor development, or a host of such differences [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] (see also the bibliography on microarray-based classification provided as part of the supplementary information to [13]). Classifiers are designed from a sample of expression vectors. This involves assessing expression levels from RNA obtained from the different tissues with microarrays, determining genes whose expression levels can be used as classifier features (variables), and then applying some rule to design the classifier from the sample microarray data. Expression values have randomness arising from both biological and experimental variability. Design, performance evaluation, and application of features must take this randomness into account. Three critical issues arise. First, given a set of variables, how does one design a classifier from the sample data that provides good classification over the general population? Second, how does one estimate the error of a designed classifier when data are limited? Third, given a large set of potential features, such as the large number of expression levels provided by each microarray, how does one select a set of features as the input to the classifier? Small samples (relative to the number of features) are ubiquitous in genomic signal processing and impact all three issues [14].

3.2. Classifier design

Classification involves a *feature vector* $\mathbf{X} = (X_1, X_2, \ldots, X_d)$ on d-dimensional Euclidean space $\mathbb{R}^d$, composed of random variables (*features*), a binary random

variable Y, and a *classifier* $\psi : \mathbb{R}^d \to \{0, 1\}$ to serve as a predictor of Y, which means that Y is to be predicted by $\psi(\mathbf{X})$. The values 0 or 1 of Y are treated as class *labels*. We assume there is a joint *feature-label distribution* F for the pair $(\mathbf{X}, Y)$ that completely characterizes the stochastic classification problem.

The space of all classifiers, which in our case is the space of all binary functions on $\mathbb{R}^d$, will be denoted by $\mathcal{F}$. The error $\varepsilon[\psi]$ of $\psi \in \mathcal{F}$ is the probability that the classification is erroneous, namely, $\varepsilon[\psi] = P(\psi(\mathbf{X}) \neq Y)$. It can be written as

$$\varepsilon[\psi] = E_F[\,|\,Y - \psi(\mathbf{X})\,|\,], \tag{3.1}$$

where the expectation is taken relative to the feature-label distribution F (as indicated by the notation E_F). In other words, $\varepsilon[\psi]$ equals the mean absolute difference between label and classification. Owing to the binary nature of $\psi(\mathbf{X})$ and Y, $\varepsilon[\psi]$ also equals the mean square error between label and classification.

3.2.1. Bayes classifier

An optimal classifier ψ_d is one having minimal error ε_d among all $\psi \in \mathcal{F}$, so that it is the minimal mean-absolute-error predictor of Y. The optimal classifier ψ_d is called the *Bayes classifier* and its error ε_d is called the *Bayes error*. The Bayes classifier, and thus the Bayes error, depends on the feature-label distribution of $(\mathbf{X}, Y)$—how well the labels are distributed among the variables being used to discriminate them, and how the variables are distributed in $\mathbb{R}^d$.

The posterior distributions for $\mathbf{X}$ are defined by $\eta_0(\mathbf{x}) = f_{\mathbf{X},Y}(\mathbf{x}, 0)/f_{\mathbf{X}}(\mathbf{x})$ and $\eta_1(\mathbf{x}) = f_{\mathbf{X},Y}(\mathbf{x}, 1)/f_{\mathbf{X}}(\mathbf{x})$, where $f_{\mathbf{X},Y}(\mathbf{x}, y)$ and $f_{\mathbf{X}}(\mathbf{x})$ are the densities for $(\mathbf{X}, Y)$ and $\mathbf{X}$, respectively. The posteriors $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ give the probability that $Y = 0$ or $Y = 1$, respectively, given $\mathbf{X} = \mathbf{x}$. Note that $\eta_0(\mathbf{x}) = 1 - \eta_1(\mathbf{x})$. Note also that, as a function of $\mathbf{X}$, $\eta_0(\mathbf{X})$ and $\eta_1(\mathbf{X})$ are random variables. Furthermore, in this binary-label setting, $\eta_1(\mathbf{x}) = E[Y|\mathbf{x}]$ is the conditional expectation of Y, given $\mathbf{x}$. The error of an arbitrary classifier can be expressed as

$$\varepsilon[\psi] = \int_{\{\mathbf{x}|\psi(\mathbf{x})=0\}} \eta_1(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} + \int_{\{\mathbf{x}|\psi(\mathbf{x})=1\}} \eta_0(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{3.2}$$

It is easy to verify that the right-hand side of (3.2) is minimized by

$$\psi_d(\mathbf{x}) = \begin{cases} 1 & \text{if } \eta_1(\mathbf{x}) \geq \eta_0(\mathbf{x}), \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

Hence, the Bayes classifier $\psi_d(\mathbf{x})$ is defined to be 1 or 0 according to whether Y is more likely to be 1 or 0, given $\mathbf{x}$ (ties may be broken arbitrarily). For this reason, the Bayes classifier is also known as the *maximum a posteriori* (MAP) classifier.

It follows from (3.2) and (3.3) that the Bayes error is given by

$$\begin{aligned}\varepsilon_d &= \int_{\{\mathbf{x}|\eta_1(\mathbf{x})<\eta_0(\mathbf{x})\}} \eta_1(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} + \int_{\{\mathbf{x}|\eta_1(\mathbf{x})\geq\eta_0(\mathbf{x})\}} \eta_0(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \\ &= E[\min\{\eta_0(\mathbf{X}), \eta_1(\mathbf{X})\}].\end{aligned} \tag{3.4}$$

By Jensen's inequality, it follows from (3.4) that $\varepsilon_d \leq \min\{E[\eta_0(\mathbf{X})], E[\eta_1(\mathbf{X})]\}$. Therefore, if either of the posteriors are uniformly small (e.g., if one of the classes is much more likely than the other), then the Bayes error is necessarily small.

The problem with the Bayes classifier is that the feature-label distribution is typically unknown, and thus so are the posteriors. Therefore, we must design a classifier from sample data. An obvious approach would be to estimate the posterior distributions from data, but often we do not have sufficient data to obtain good estimates. Moreover, good classifiers can be obtained even when we lack sufficient data for satisfactory density estimation.

3.2.2. Classification rules

Design of a classifier ψ_n from a random sample $S_n = \{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_n, Y_n)\}$ of vector-label pairs drawn from the feature-label distribution requires a classification rule that operates on random samples to yield a classifier. A *classification rule* is a mapping $\Psi_n : [\mathbb{R}^d \times \{0,1\}]^n \to \mathcal{F}$. Given a sample S_n, we obtain a designed classifier $\psi_n = \Psi_n(S_n) \in \mathcal{F}$, according to the rule Ψ_n. To be fully formal, one might write $\psi_n(S_n; \mathbf{X})$ rather than $\psi_n(\mathbf{X})$; however, we will use the simpler notation, keeping in mind that ψ_n derives from a classification rule applied to a feature-label sample. Note that what is usually called a classification rule is really a sequence of classification rules depending on n. Figure 3.1 presents an example of a linear designed classifier, obtained via the linear-discriminant-analysis (LDA) classification rule (see Section 3.2.4.9). The sample data in this example consist of expression values of two top discriminatory genes on a total of 295 microarrays from a cancer classification study [15] (see Section 3.2.5 for more details about this data set).

The Bayes error ε_d is estimated by the expected error of the designed classifier $\varepsilon_n = \varepsilon[\psi_n]$. There is a *design error*

$$\Delta_n = \varepsilon_n - \varepsilon_d, \tag{3.5}$$

ε_n and Δ_n being sample-dependent random variables. The expected design error is $E[\Delta_n]$, the expectation being relative to all possible samples. The expected error of ψ_n is decomposed according to

$$E[\varepsilon_n] = \varepsilon_d + E[\Delta_n]. \tag{3.6}$$

The quantity $E[\varepsilon_n]$, or alternatively $E[\Delta_n]$, measures the global properties of classifications rules, rather than the performance of classifiers designed on individual

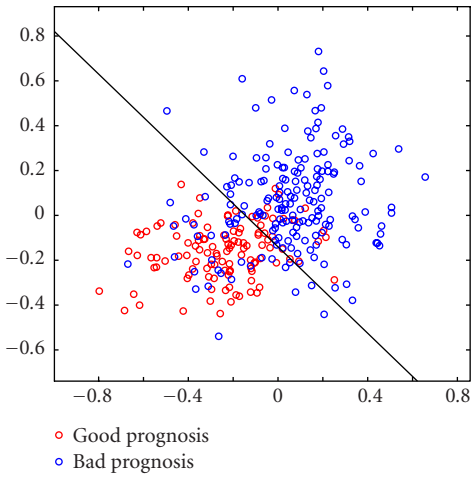


Figure 3.1. Example of a linear designed classifier.

samples (on the other hand, a classification rule for which $E[\varepsilon_n]$ is small will also tend to produce designed classifiers that display small error).

Asymptotic properties of a classification rule concern large samples (as $n \to \infty$). A rule is said to be *consistent* for a feature-label distribution of $(\mathbf{X}, Y)$ if $\Delta_n \to 0$ in the mean, meaning $E[\Delta_n] \to 0$ as $n \to \infty$. For a consistent rule, the expected design error can be made arbitrarily small for a sufficiently large amount of data. Since the feature-label distribution is unknown a priori, rules for which convergence is independent of the distribution are desirable. A classification rule is *universally consistent* if $\Delta_n \to 0$ in the mean for any distribution of $(\mathbf{X}, Y)$. Universal consistency is useful for large samples, but has little consequence for small samples.

3.2.3. Constrained classifier design

A classification rule can yield a classifier that makes very few, or no, errors on the sample data on which it is designed, but performs poorly on the distribution as a whole, and therefore on new data to which it is applied. This situation is exacerbated by complex classifiers and small samples. If the sample size is dictated by experimental conditions, such as cost or the availability of patient RNA for expression microarrays, then one only has control over classifier complexity. The situation with which we are concerned is typically referred to as *overfitting*. The basic point is that a classification rule should not cut up the space in a manner too complex for the amount of sample data available. This might improve the *apparent* error rate (i.e., the number of errors committed by the classifier using the training data as testing points), but at the same time it will most likely worsen the true error of the classifier for independent future data (also called the *generalization error* in this context). The problem is not necessarily mitigated by applying an

error-estimation rule—perhaps more sophisticated than the apparent error rate—to the designed classifier to see if it "actually" performs well, since when there is only a small amount of data available, error-estimation rules are very imprecise (as we will see in Section 3.4), and the imprecision tends to be worse for complex classification rules. Hence, a low error estimate is not sufficient to overcome our expectation of a large expected error when using a complex classifier with a small data set. Depending on the amount of data available, we need to consider constrained classification rules.

Constraining classifier design means restricting the functions from which a classifier can be chosen to a class $\mathcal{C} \subseteq \mathcal{F}$. This leads to trying to find an optimal *constrained classifier* $\psi_{\mathcal{C}} \in \mathcal{C}$ having error $\varepsilon_{\mathcal{C}}$. Constraining the classifier can reduce the expected error, but at the cost of increasing the error of the best possible classifier. Since optimization in $\mathcal{C}$ is over a subclass of classifiers, the error $\varepsilon_{\mathcal{C}}$ of $\psi_{\mathcal{C}}$ will typically exceed the Bayes error, unless the Bayes classifier happens to be in $\mathcal{C}$. This cost of constraint (approximation) is

$$\Delta_{\mathcal{C}} = \varepsilon_{\mathcal{C}} - \varepsilon_d. \tag{3.7}$$

A classification rule yields a classifier $\psi_{n,\mathcal{C}} \in \mathcal{C}$, with error $\varepsilon_{n,\mathcal{C}}$, and $\varepsilon_{n,\mathcal{C}} \geq \varepsilon_{\mathcal{C}} \geq \varepsilon_d$. Design error for constrained classification is

$$\Delta_{n,\mathcal{C}} = \varepsilon_{n,\mathcal{C}} - \varepsilon_{\mathcal{C}}. \tag{3.8}$$

For small samples, this can be substantially less than Δ_n, depending on $\mathcal{C}$ and the classification rule. The error of the designed constrained classifier is decomposed as

$$\varepsilon_{n,\mathcal{C}} = \varepsilon_d + \Delta_{\mathcal{C}} + \Delta_{n,\mathcal{C}}. \tag{3.9}$$

Therefore, the expected error of the designed classifier from $\mathcal{C}$ can be decomposed as

$$E\left[\varepsilon_{n,\mathcal{C}}\right] = \varepsilon_d + \Delta_{\mathcal{C}} + E\left[\Delta_{n,\mathcal{C}}\right]. \tag{3.10}$$

The constraint is beneficial if and only if $E[\varepsilon_{n,\mathcal{C}}] < E[\varepsilon_n]$, that is, if

$$\Delta_{\mathcal{C}} < E\left[\Delta_n\right] - E\left[\Delta_{n,\mathcal{C}}\right]. \tag{3.11}$$

If the cost of a constraint is less than the decrease in expected design error, then the expected error of $\psi_{n,\mathcal{C}}$ is less than that of ψ_n. The dilemma is as follows: strong constraint reduces $E[\Delta_{n,\mathcal{C}}]$ at the cost of increasing $\varepsilon_{\mathcal{C}}$.

The matter can be graphically illustrated. For two classification rules to be shortly introduced, the discrete-data plug-in rule and the cubic histogram rule with fixed cube size, $E[\Delta_n]$ is nonincreasing, meaning that $E[\Delta_{n+1}] \leq E[\Delta_n]$. This means that the expected design error never increases as sample sizes increase, and

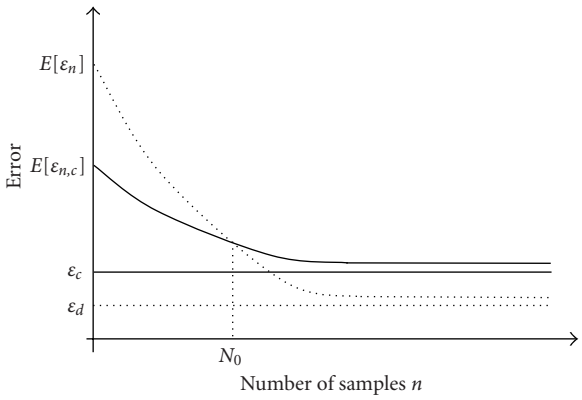


Figure 3.2. Errors of unconstrained and constrained classifiers.

it holds for any feature-label distribution. Such classification rules are called *smart*. They fit our intuition about increasing sample sizes. Now consider a consistent rule, constraint, and distribution for which $E[\Delta_{n+1}] \leq E[\Delta_n]$ and $E[\Delta_{n+1,\mathcal{C}}] \leq E[\Delta_{n,\mathcal{C}}]$. Figure 3.2 illustrates the design problem. If n is sufficiently large, then $E[\varepsilon_n] < E[\varepsilon_{n,\mathcal{C}}]$; however, if n is sufficiently small, then $E[\varepsilon_n] > E[\varepsilon_{n,\mathcal{C}}]$. The point N_0 at which the decreasing lines cross is the cutoff: for $n > N_0$, the constraint is detrimental; for $n < N_0$, it is beneficial. When $n < N_0$, the advantage of the constraint is the difference between the decreasing solid and dashed curves.

A fundamental theorem provides bounds for $E[\Delta_{n,\mathcal{C}}]$ [16]. The *empirical-error rule* chooses the classifier in $\mathcal{C}$ that makes the least number of errors on the sample data. For this (intuitive) rule, $E[\Delta_{n,\mathcal{C}}]$ satisfies the bound

$$E[\Delta_{n,\mathcal{C}}] \leq 8\sqrt{\frac{V_{\mathcal{C}} \log n + 4}{2n}}, \tag{3.12}$$

where $V_{\mathcal{C}}$ is the *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{C}$. Details of the VC dimension are outside the scope of this paper. Nonetheless, it is clear from (3.12) that n must greatly exceed $V_{\mathcal{C}}$ for the bound to be small.

To illustrate the problematic nature of complex (high-VC-dimension) classifiers, we apply the preceding bound to two classifier classes to be introduced in the next section. The VC dimension of a linear classifier is $d + 1$, where d is the number of variables, whereas the VC dimension of a neural network (NNET) with an even number k of neurons has the lower bound $V_{\mathcal{C}} \geq dk$ [17]. If k is odd, then $V_{\mathcal{C}} \geq d(k-1)$. Thus, if one wants to use a large number of neurons to obtain a classifier that can very finely fit the data, the VC dimension can greatly exceed that of a linear classifier. To appreciate the implications, suppose $d = k = 10$. Then the VC dimension of a NNET is bounded below by 100. Setting $V_{\mathcal{C}} = 100$ and $n = 5000$ in (3.12) yields a bound exceeding 1, which says nothing. Not only is the inequality in (3.12) a bound, it is worst case because there are no distributional assumptions. The situation may not be nearly so bad. Still, one must proceed with

care, especially in the absence of distributional knowledge. Increasing complexity is often counterproductive unless there is a large sample available. Otherwise, one could easily end up with a very bad classifier whose error estimate is very small!

3.2.4. Specific classification rules

In this section of the chapter, we discuss some commonly employed classification rules, beginning with a rule that is employed in different manners to produce related rules.

3.2.4.1. Plug-in rule

Considering the Bayes classifier defined by (3.3), let $\eta_{1,n}(\mathbf{x})$ be an estimate of $\eta_1(\mathbf{x})$ based on a sample S_n, and let $\eta_{0,n}(\mathbf{x}) = 1 - \eta_{1,n}(\mathbf{x})$. A reasonable classification rule is to define $\psi_n(\mathbf{x})$ according to (3.3) with $\eta_{0,n}(\mathbf{x})$ and $\eta_{1,n}(\mathbf{x})$ in place of $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$, respectively. For this *plug-in rule*,

$$\Delta_n = \int_{\{\mathbf{x}:\psi_n(\mathbf{x})\neq\psi_d(\mathbf{x})\}} \left|\eta_{1,n}(\mathbf{x}) - \eta_{0,n}(\mathbf{x})\right| f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}. \tag{3.13}$$

A sufficient condition for the plug-in rule to be consistent is given by [18]:

$$\lim_{n\to\infty} \int_{\mathbb{R}^d} \left|\eta_1(\mathbf{x}) - \eta_{1,n}(\mathbf{x})\right|^{1/2} d\mathbf{x} = 0. \tag{3.14}$$

3.2.4.2. Histogram rule

Suppose that $\mathbb{R}^d$ is partitioned into cubes of equal side length r_n. For each point $\mathbf{x} \in \mathbb{R}^d$, the *histogram rule* defines $\psi_n(\mathbf{x})$ to be 0 or 1 according to which is the majority among the labels for points in the cube containing $\mathbf{x}$. If the cubes are defined so that $r_n \to 0$ and $nr_n^d \to \infty$ as $n \to \infty$, then the rule is universally consistent [19].

3.2.4.3. Multinomial discrimination

The situation in which only a finite number of observed patterns are possible, say $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m$, is referred to as *multinomial discrimination*. An important example is the so-called *fundamental rule* [18], which assigns at each pattern $\mathbf{z}_i$ the label with the maximum relative frequency among all sample points corresponding to $\mathbf{z}_i$. It can be checked easily that this is the plug-in version of (3.3)—for this reason, the fundamental rule is also called the *discrete-data plug-in rule*. The fundamental rule corresponds to a special case of the histogram rule, when the partition used is such that each cell contains exactly one of the possible patterns. For a zero Bayes error and equiprobable patterns, we have that $E[\varepsilon_n] \geq (1 - 1/m)^n$, which shows clearly the effect of using too small a sample. Indeed, if $n \leq m/2$, then the inequality yields $E[\varepsilon_n] \geq 0.5$, which shows that the fundamental rule is useless in this

case. In the other direction (for large samples), it is shown in [18] that the fundamental rule is universally consistent and $E[\varepsilon_n] \leq \varepsilon_d + 1.075\sqrt{m/n}$. Multinomial discrimination plays a key role in gene prediction for quantized expression values, in particular, binarized gene expressions in which a gene is qualitatively labeled as ON (1) or OFF (0) [20, 21, 22]. In this situation, if there are r binary gene values used to predict a target gene value, then $m = 2^r$ and prediction reduces to multinomial discrimination. Extension to the case of any finite expression quantization is straightforward. This kind of quantization occurs with discrete gene regulatory networks, in particular, Boolean networks [23, 24]. In a related vein, it has been shown that binarized (ON, OFF) expression values can be used to obtain good classification [25] and clustering [26].

3.2.4.4. k-nearest-neighbor rule

For the basic *nearest-neighbor (NN) rule* , ψ_n is defined for each $\mathbf{x} \in \mathbb{R}^d$ by letting $\psi_n(\mathbf{x})$ take the label of the sample point closest to $\mathbf{x}$. For the NN rule, no matter the feature-label distribution of $(\mathbf{X}, Y)$, $\varepsilon_d \leq \lim_{n\to\infty} E[\varepsilon_n] \leq 2\varepsilon_d$ [27]. It follows that $\lim_{n\to\infty} E[\Delta_n] \leq \varepsilon_d$. Hence, the asymptotic expected design error is small if the Bayes error is small; however, this result does not give consistency. More generally, for the *k-nearest-neighbor rule (kNN)*, with k odd, the k points closest to $\mathbf{x}$ are selected and $\psi_n(\mathbf{x})$ is defined to be 0 or 1 according to which is the majority among the labels of these points. If $k = 1$, this gives the NN rule. The limit of $E[\varepsilon_n]$ as $n \to \infty$ can be expressed analytically and various upper bounds exist. In particular, $\lim_{n\to\infty} E[\Delta_n] \leq (ke)^{-1/2}$. This does not give consistency, but it does show that the design error gets arbitrarily small for sufficiently large k as $n \to \infty$. The kNN rule is universally consistent if $k \to \infty$ and $k/n \to 0$ as $n \to \infty$ [28].

3.2.4.5. Kernel rules

The *moving-window rule* takes the majority label among all sample points within a specified distance of $\mathbf{x}$. The rule can be "smoothed" by giving weights to different sample points: the weights associated with the 0- and 1-labeled sample points are added up separately, and the output is defined to be the label with the larger sum. A *kernel rule* is constructed by defining a weighting kernel based on the distance of a sample point from $\mathbf{x}$. The *Gaussian kernel* is defined by $K_h(\mathbf{x}) = e^{-\|\mathbf{x}/h\|^2}$, whereas the *Epanechnikov kernel* is given by $K_h(\mathbf{x}) = 1 - \|\mathbf{x}/h\|^2$ if $\|\mathbf{x}\| \leq h$ and $K_h(\mathbf{x}) = 0$ if $\|\mathbf{x}\| > h$. If $\mathbf{x}$ is the point at which the classifier is being defined, then the weight at a sample point $\mathbf{x}_k$ is $K_h(\mathbf{x}-\mathbf{x}_k)$. Since the Gaussian kernel is never 0, all sample points get some weight. The Epanechnikov kernel is 0 for sample points at a distance more than h from $\mathbf{x}$, so that, like the moving-window rule, only sample points within a certain radius contribute to the definition of $\psi_n(\mathbf{x})$. The moving-window rule is a special case of a kernel rule with the weights being 1 within a specified radius. The kernel rules we have given are universally consistent [18].

3.2.4.6. Linear classifiers

For classification rules determined by parametric representation, the classifier is postulated to have a functional form $\psi(x_1, x_2, \ldots, x_d; a_0, a_1, \ldots, a_r)$, where the parameters $a_0, a_1, \ldots, a_r$ are to be determined by some estimation procedure based on the sample data. For parametric representation, we assume the labels to be -1 and 1. The most basic functional form involves a linear combination of the coordinates of the observations. A binary function is obtained by thresholding. A *linear classifier*, or *perceptron*, has the form

$$\psi(x) = \mathrm{T}\left[a_0 + \sum_{i=1}^{d} a_i x_i\right], \tag{3.15}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ and T thresholds at 0 and yields -1 or 1. A linear classifier divides the space into two half spaces determined by the hyperplane defined by the parameters $a_0, a_1, \ldots, a_d$. The hyperplane is determined by the equation formed from setting the linear combination equal to 0. Using the dot product $\mathbf{a} \cdot \mathbf{x}$, which is equal to the sum in the preceding equation absent the constant term a_0, the hyperplane is defined by $\mathbf{a} \cdot \mathbf{x} = -a_0$. Numerous design procedures have been proposed to avoid the computational requirement of full optimization for linear classifiers. Each finds parameters that hopefully define a linear classifier whose error is close to optimal. Often, analysis of the design procedure depends on whether the sample data are *linearly separable*, meaning there exists a hyperplane such that points with label -1 lie on one side of the hyperplane and the points with label 1 lie on the other side. There are many design algorithms for linear classification, each meant to achieve some advantage relative to other methods.

3.2.4.7. Support vector machines

The *support vector machine* (*SVM*) provides a method for designing linear classifiers [29]. Figure 3.3 shows a linearly separable data set and three hyperplanes (lines). The outer lines pass through points in the sample data, and the third, called the *maximal-margin hyperplane* (*MMH*) is equidistant between the outer lines. It has the property that the distance from it to the nearest -1-labeled sample point is equal to the distance from it to the nearest 1-labeled sample point. The sample points closest to it are called *support vectors* (the circled sample points in Figure 3.3). The distance from the MMH to any support vector is called the *margin*. The matter is formalized by recognizing that differently labeled sets are separable by the hyperplane $\mathbf{u} \cdot \mathbf{x} = c$, where $\mathbf{u}$ is a unit vector and c is a constant, if $\mathbf{u} \cdot \mathbf{x}_k > c$ for $y_k = 1$ and $\mathbf{u} \cdot \mathbf{x}_k < c$ for $y_k = -1$. For any unit vector $\mathbf{u}$, the margin is given by

$$\rho(u) = \frac{1}{2}\left(\min_{\{\mathbf{x}^k : y^k = 1\}} \mathbf{u} \cdot \mathbf{x}_k - \max_{\{\mathbf{x}^k : y^k = -1\}} \mathbf{u} \cdot \mathbf{x}_k\right). \tag{3.16}$$

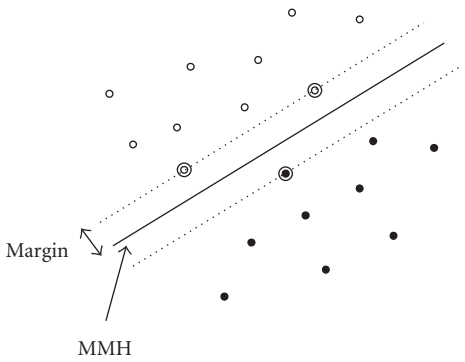


Figure 3.3. MMH for linearly separable data. The support vectors are the circled sample points.

The MMH, which is unique, can be found by solving the following quadratic optimization problem:

$$\begin{aligned} &\min \|\mathbf{v}\|, \quad \text{subject to} \\ &\mathbf{v}\cdot\mathbf{x}_k + b \geq 1 \quad \text{if } y_k = 1, \\ &\mathbf{v}\cdot\mathbf{x}_k + b \leq -1 \quad \text{if } y_k = -1. \end{aligned} \tag{3.17}$$

If $\mathbf{v}_0$ satisfies this optimization problem, then the vector defining the MMH and the margin are given by $\mathbf{u}_0 = \mathbf{v}_0/\|\mathbf{v}_0\|$ and $\rho(\mathbf{u}_0) = \|\mathbf{v}_0\|^{-1}$, respectively.

If the sample is not linearly separable, then one has two choices: find a reasonable linear classifier or find a nonlinear classifier. In the first case, the preceding method can be modified by making appropriate changes to the optimization problem (3.17); in the second case, one can map the sample points into a higher-dimensional space where they are linearly separable, find a hyperplane in that space, and then map back into the original space (we refer the reader to [29] for details).

3.2.4.8. Quadratic discriminant analysis

Let R_k denote the region in $\mathbb{R}^d$, where the Bayes classifier has the value k, for $k = 0, 1$. According to (3.3), $\mathbf{x} \in R_k$ if $\eta_k(\mathbf{x}) > \eta_j(\mathbf{x})$, for $j \neq k$ (ties in the posteriors being broken arbitrarily). Since $\eta_k(\mathbf{x}) = f_{\mathbf{X}|Y}(\mathbf{x}|k)f_Y(k)/f_{\mathbf{X}}(\mathbf{x})$, upon taking the logarithm and discarding the common term $f_{\mathbf{X}}(\mathbf{x})$, this is equivalent to $\mathbf{x} \in R_k$ if $d_k(\mathbf{x}) > d_j(\mathbf{x})$, where the *discriminant* $d_k(\mathbf{x})$ is defined by

$$d_k(\mathbf{x}) = \log f_{\mathbf{X}|Y}(\mathbf{x}|k) + \log f_Y(k). \tag{3.18}$$

If the conditional densities $f_{\mathbf{X}|Y}(\mathbf{x}|0)$ and $f_{\mathbf{X}|Y}(\mathbf{x}|1)$ are normally distributed, then

$$f_{\mathbf{X}|Y}(\mathbf{x}|k) = \frac{1}{\sqrt{(2\pi)^n \det[\mathbf{K}_k]}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{u}_k)'\mathbf{K}_k^{-1}(\mathbf{x}-\mathbf{u}_k) \right], \tag{3.19}$$

where $\mathbf{K}_k$ and $\mathbf{u}_k$ are the covariance matrix and mean vector for class k, respectively. Dropping the constant terms and multiplying by the factor 2 (which has no effect on classification), the discriminant becomes

$$d_k(\mathbf{x}) = -(\mathbf{x} - \mathbf{u}_k)' \mathbf{K}_k^{-1} (\mathbf{x} - \mathbf{u}_k) - \log\left(\det\left[\mathbf{K}_k\right]\right) + 2\log f_Y(k). \tag{3.20}$$

Hence, the discriminant is quadratic in $\mathbf{x}$. The first term in (3.20) is known as the *Mahalanobis distance* between $\mathbf{x}$ and $\mathbf{u}_k$. A simple calculation shows that the optimal decision boundary $d_1(\mathbf{x}) - d_0(\mathbf{x}) = 0$ is given by

$$\begin{aligned}\mathbf{x}'(\mathbf{K}_0^{-1} - \mathbf{K}_1^{-1})\mathbf{x} - 2(\mathbf{u}_0'\mathbf{K}_0^{-1} - \mathbf{u}_1'\mathbf{K}_1^{-1})\mathbf{x} + \mathbf{u}_0'\mathbf{K}_0^{-1}\mathbf{u}_0 - \mathbf{u}_1'\mathbf{K}_1^{-1}\mathbf{u}_1 \\ + \log\left(\frac{\det\left[\mathbf{K}_0\right]}{\det\left[\mathbf{K}_1\right]}\right) + 2\log\left(\frac{f_Y(1)}{f_Y(0)}\right) = 0.\end{aligned} \tag{3.21}$$

This is an equation in the form $\mathbf{x}'\mathbf{A}\mathbf{x} - \mathbf{b}'\mathbf{x} + c = 0$. In 2-dimensional space, such an equation produces *conical-section* decision curves, whereas in 3-dimensional spaces, it produces decision surfaces known as *quadrics*. Plugging sample-based estimates for the covariance matrices, mean vectors, and priors into (3.21), leads to a classification rule known as *quadratic discriminant analysis* (*QDA*). Depending on the estimated coefficients in (3.21), decision boundaries ranging from paraboloids to spheres can be produced by QDA.

3.2.4.9. Linear discriminant analysis

If both conditional densities possess the same covariance matrix $\mathbf{K}$, then the quadratic term and the first logarithmic term vanish in (3.21), yielding

$$(\mathbf{u}_1 - \mathbf{u}_0)'\mathbf{K}^{-1}\mathbf{x} - \frac{1}{2}(\mathbf{u}_1'\mathbf{K}^{-1}\mathbf{u}_1 - \mathbf{u}_0'\mathbf{K}^{-1}\mathbf{u}_0) + \log\left(\frac{f_Y(1)}{f_Y(0)}\right) = 0. \tag{3.22}$$

This is an equation in the form $\mathbf{a}\mathbf{x}' + m = 0$. Such equations produce decision surfaces that are hyperplanes in d-dimensional space. Plugging into (3.22) sample-based estimates for the covariance matrix, mean vectors, and priors leads to a classification rule known as *linear discriminant analysis* (*LDA*). In practice, the usual maximum-likelihood estimates are employed for the mean vectors, whereas the estimate of the covariance matrix is often given by the *pooled covariance matrix*:

$$\hat{\mathbf{K}} = \frac{1}{2}(\hat{\mathbf{K}}_0 + \hat{\mathbf{K}}_1), \tag{3.23}$$

where $\hat{\mathbf{K}}_k$ is the usual maximum-likelihood estimate of the covariance matrix of class k (note that, in general, $\hat{\mathbf{K}}_0 \neq \hat{\mathbf{K}}_1$). In addition, especially in the case of small sample sizes, it is a common practice to assume equally likely classes, so that the term $\log(f_Y(1)/f_Y(0))$ is zero. This avoids the use of unreliable estimates of the priors derived from limited data.

3.2.4.10. Nearest-mean classifier

If, besides a common covariance matrix and equally likely classes one assumes uncorrelated conditional distributions, with covariance matrix $\mathbf{K} = \sigma^2\mathbf{I}$, then (3.22) reduces to

$$(\mathbf{u}_1 - \mathbf{u}_0)'\mathbf{x} - \frac{1}{2}\left(||\mathbf{u}_1||^2 - ||\mathbf{u}_0||^2\right) = 0. \tag{3.24}$$

The optimal hyperplane in this case is perpendicular to the line joining the means and passes through the midpoint of that line. Therefore, a sample point is assigned to class k if its distance to the mean vector $\mathbf{u}_k$ is minimal. This also follows from the fact that the discriminant function in (3.20) can be written in this case simply as $d_k(\mathbf{x}) = -\|\mathbf{x} - \mathbf{u}_k\|$. Substituting sample-based mean estimates for the mean vectors in (3.24) leads to the *nearest-mean classifier* (*NMC*). This classification rule has the advantage of avoiding the estimation (and inversion) of the covariance matrices, so it can be effective in extreme small-sample scenarios.

Equations (3.21), (3.22), and (3.24), for the QDA, LDA, and NMC rules, respectively, were derived under the Gaussian assumption, but in practice can perform well so long as the underlying class-conditional densities are approximately Gaussian—and one can obtain good estimates of the relevant covariance matrices. Owing to the greater number of parameters to be estimated for QDA as opposed to LDA and NMC, one can proceed with smaller samples for LDA than with QDA, and in extreme small-sample cases, NMC may be the most effective choice, due to its avoiding the estimation of the covariance matrices. Of course, if the assumption of equal and/or uncorrelated covariance matrices does not hold, then the LDA and NMC rules will have asymptotic expected error biased away from the Bayes error. Therefore, in large-sample scenarios, QDA is preferable. However, LDA has been reported to be more robust relative to the underlying Gaussian assumption than QDA [30]. In our experience, LDA has proved to be a very robust classification rule (see Section 3.2.5), which is effective for a wide range of sample sizes.

3.2.4.11. Neural networks

A (feed-forward) two-layer *neural network* has the form

$$\psi(\mathbf{x}) = \mathrm{T}\left[c_0 + \sum_{i=1}^{k} c_i\sigma\left[\psi_i(\mathbf{x})\right]\right], \tag{3.25}$$

where T thresholds at 0, σ is a *sigmoid function* (i.e., a nondecreasing function with limits -1 and $+1$ at $-\infty$ and ∞, resp.), and

$$\psi_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{d} a_{ij}x_j. \tag{3.26}$$

Each operator in the sum of (3.25) is called a *neuron*. These form the hidden layer. We consider NNETs with the threshold sigmoid: $\sigma(x) = -1$ if $x \leq 0$ and $\sigma(x) = 1$ if $x > 0$. If $k \to \infty$ such that $(k \log n)/n \to 0$ as $n \to \infty$, then, as a class, NNETs are universally consistent [31], but one should beware of the increasing number of neurons required.

A key point here is that any function whose absolute value possesses finite integral can be approximated arbitrarily closely by a sufficiently complex NNET. While this is theoretically important, there are limitations to its practical usefulness. Not only does one not know the function, in this case the Bayes classifier, whose approximation is desired, but even were we to know the function and how to find the necessary coefficients, a close approximation can require an extremely large number of model parameters. Given the NNET structure, the task is to estimate the optimal weights. As the number of model parameters grows, the use of the model for classifier design becomes increasingly intractable owing to the increasing amount of data required for estimation of the model parameters. Since the number of hidden units must be kept relatively small, thereby requiring significant constraint, when data are limited, there is no assurance that the optimal NNET of the prescribed form closely approximates the Bayes classifier. Model estimation is typically done by some iterative procedure, with advantages and disadvantages being associated with different methods [32].

3.2.4.12. Classification trees

The histogram rule partitions the space without reference to the actual data. One can instead partition the space based on the data, either with or without reference to the labels. Tree classifiers are a common way of performing data-dependent partitioning. Since any tree can be transformed into a binary tree, we only need to consider binary classification trees. A tree is constructed recursively based on some criteria. If S represents the set of all data, then it is partitioned according to some rule into $S = S_1 \cup S_2$. There are four possibilities: (i) S_1 is partitioned into $S_1 = S_{11} \cup S_{12}$ and S_2 is partitioned into $S_2 = S_{21} \cup S_{22}$; (ii) S_1 is partitioned into $S_1 = S_{11} \cup S_{12}$ and partitioning of S_2 is terminated; (iii) S_2 is partitioned into $S_2 = S_{21} \cup S_{22}$ and partitioning of S_1 is terminated; and (iv) partitioning of both S_1 and S_2 is terminated. In the last case, the partition is complete; in any of the others, it proceeds recursively until all branches end in termination, at which point the leaves on the tree represent the partition of the space. On each cell (subset) in the final partition, the designed classifier is defined to be 0 or 1, according to which is the majority among the labels of the points in the cell.

A wide variety of classification trees, whose leaves are rectangles in $\mathbb{R}^d$, can be obtained by perpendicular splits. At each stage of growing the tree, a decision to split a rectangle R is made according to a coordinate decision of the form $x_i^j \leq \alpha$, where $\mathbf{x}^j = (x_1^j, x_2^j, \ldots, x_d^j)$ is a sample point in $\mathbb{R}^d$. Also at each stage, there are two collections of rectangles, R_0 and R_1, determined by majority vote of the labels, so that $R \in R_1$ if and only if the majority of labels for points in R have value 1. The 0 and 1 decision regions are determined by the unions of rectangles in

R_0 and R_1, respectively. A final classification tree, and therefore the designed classifier, depends on the splitting criterion, choice of α, and a stopping criterion. Two desirable attributes of a stopping criterion are that the leaf nodes (final rectangles) be small in number so that the complexity of the classifier be not too great for the amount of data (thus avoiding overfitting), and that the labels in each final rectangle be not evenly split, thereby increasing the likelihood that the majority label accurately reflects the distribution in the rectangle. A rectangle is said to be *pure* relative to a particular sample if all labels corresponding to points in the rectangle possess the same label.

One popular method of splitting, which goes under the name *classification and regression trees* (*CART*), is based on the notion of an *impurity function*. For any rectangle R, let $N_0(R)$ and $N_1(R)$ be the numbers of 0-labeled and 1-labeled points, respectively, in R, and let $N(R) = N_0(R) + N_1(R)$ be the total number of points in R. The *impurity* of R is defined by

$$\kappa(R) = \xi(p_R, 1 - p_R), \tag{3.27}$$

where $p_R = N_0(R)/N(R)$ is the proportion of 0 labels in R, and where $\xi(p, 1-p)$ is a nonnegative function satisfying the following conditions: (1) $\xi(0.5, 0.5) \geq \xi(p, 1-p)$ for any $p \in [0,1]$; (2) $\xi(0,1) = \xi(1,0) = 0$; and (3) as a function of p, $\xi(p, 1-p)$ increases for $p \in [0, 0.5]$ and decreases for $p \in [0.5, 1]$. Several observations follow from the definition of ξ: (1) $\kappa(R)$ is maximum when the proportions of 0-labeled and 1-labeled points in R are equal (corresponding to maximum impurity); (2) $\kappa(R) = 0$ if R is pure; and (3) $\kappa(R)$ increases for greater impurity.

We mention three possible choices for ξ:

(1) $\xi_e(p, 1-p) = -p \log p - (1-p)\log(1-p)$ (*entropy impurity*);
(2) $\xi_g(p, 1-p) = p(1-p)$ (*Gini impurity*);
(3) $\xi_m(p, 1-p) = \min(p, 1-p)$ (*misclassification impurity*).

The origins of these three impurities lie in the definition of $\kappa(R)$: $\xi_e(p, 1-p)$ provides an entropy estimate; $\xi_g(p, 1-p)$ provides a variance estimate for a binomial distribution; and $\xi_m(p, 1-p)$ provides an error-rate estimate.

A splitting regimen is determined by the manner in which a split will cause an overall decrease in impurity. Let i be a coordinate, α be a real number, R be a rectangle to be split along the ith coordinate, $R^i_{\alpha,-}$ be the subrectangle resulting from the ith coordinate being less than or equal to α, and $R^i_{\alpha,+}$ be the subrectangle resulting from the ith coordinate being greater than α. Define the *impurity decrement* by

$$\Delta_i(R, \alpha) = \kappa(R) - \frac{N(R^i_{\alpha,-})}{N(R)}\kappa(R^i_{\alpha,-}) - \frac{N(R^i_{\alpha,+})}{N(R)}\kappa(R^i_{\alpha,+}). \tag{3.28}$$

A good split will result in impurity reductions in the subrectangles. In computing $\Delta_i(R, \alpha)$, the new impurities are weighted by the proportions of points going into the subrectangles. CART proceeds iteratively by splitting a rectangle at $\hat{\alpha}$ on the

ith coordinate if $\Delta_i(R, \alpha)$ is maximized for $\alpha = \hat{\alpha}$. There are two possible splitting strategies: (i) the coordinate i is given and $\Delta_i(R, \alpha)$ is maximized over all α and R; (ii) the coordinate is not given and $\Delta_i(R, \alpha)$ is maximized over all i, α, and R. Various stopping strategies are possible—for instance, stopping when maximization of $\Delta_i(R, \alpha)$ yields a value below a preset threshold, or when there are fewer than a specified number of sample points assigned to the node. One may also continue to grow the tree until all leaves are pure and then prune.

3.2.5. Classification performance

In this section, we present classification results obtained with real patient data. Our purpose is to compare the performance of several of the classification rules described in the previous sections, in terms of the expected classification error, for different sample sizes and number of variables (dimensionality).

The data used in the experiments come from a microarray-based classification study [15] that analyzed a large number of microarrays, prepared with RNA from breast tumor samples from each of 295 patients (see Figure 3.1 for a plot of the expression values of two genes in these data). Using a previously established 70-gene prognosis profile [33], a prognosis signature based on gene expression was proposed in [15], which correlated well with patient survival data and other existing clinical measures. Of the $N = 295$ microarrays, $N_0 = 115$ belong to the "good-prognosis" class, whereas the remaining $N_1 = 180$ belong to the "poor-prognosis" class.

Our experiments were set up in the following way. We used log-ratio gene expression values associated with the top genes found in [33]. We consider four basic cases, corresponding to $d = 2, 3, 4, 5$ genes. In each case, we searched the best combination, in terms of estimated Bayes error, of d genes among the top 10 genes, with the purpose of not considering situations where there is too much confusion between the classes, which makes the expected errors excessively large. The Bayes error was computed by using (3.4) in conjunction with a Gaussian-kernel density estimation method, for which the kernel variance is automatically selected by a pseudolikelihood-based technique [34].

In each case, 1000 observations S_n of size ranging from $n = 20$ to $n = 120$, in steps of 5, were drawn independently from the pool of 295 microarrays. Sampling was stratified in the sense that the proportion of each class in the random sample was fixed to N_0/N for the first class and N_1/N for the second class. A classifier was designed for each sample S_n, using one of the classification rules described previously, and its classification error was approximated by means of a holdout estimator (see Section 3.4.1), whereby the $295 - n$ sample points not drawn are used as an independent test set (this is a good approximation to the true error, given the large test sample). The errors for the 1000 independent sample sets were averaged to provide a Monte Carlo estimate of the expected error for the classification rule.

Figure 3.4 displays four plots, one for each dimensionality considered. We have considered seven classification rules: LDA, QDA, NMC, 1-nearest neighbor

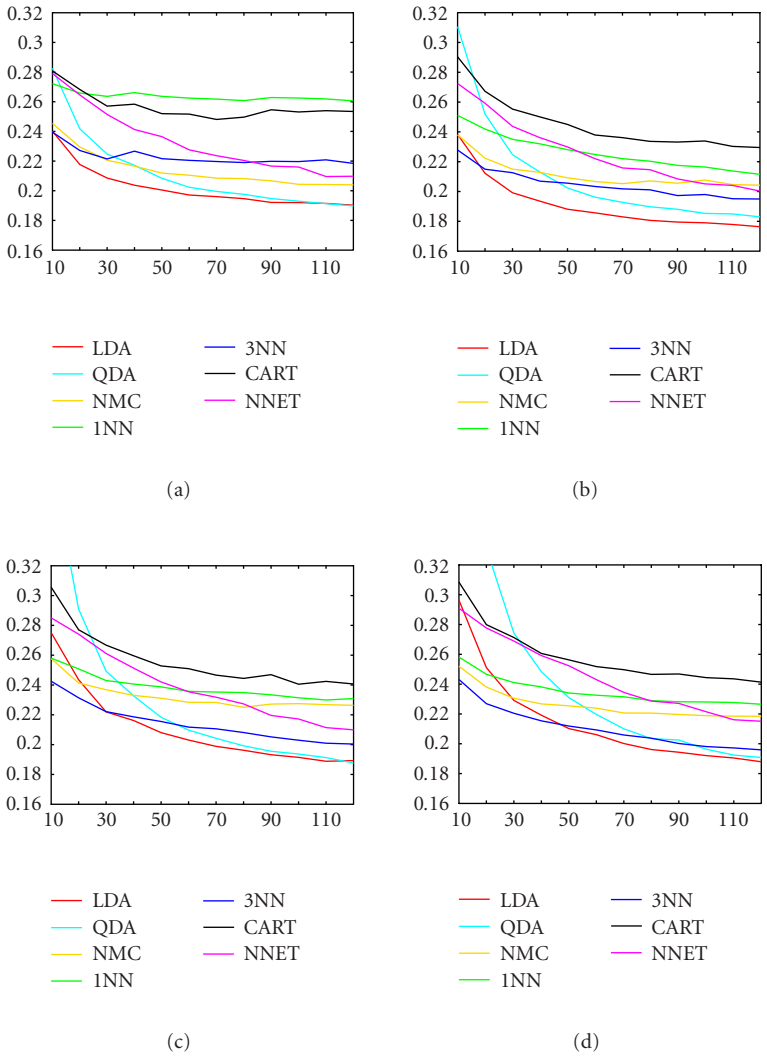


Figure 3.4. Expected error versus sample size for several classification rules and number of genes: (a) 2 genes, (b) 3 genes, (c) 4 genes, and (d) 5 genes.

(1NN), 3-nearest neighbor (3NN), CART with a stopping rule that ends splitting when there are six or fewer sample points in a node, and an NNET with 4 nodes in the hidden layer.

Confirming observations we have made previously, we can see that LDA performs quite well, and so does 3NN. We see that QDA does a very good job for larger sample sizes, but its performance degrades quickly for smaller sample sizes.

NMC does a very credible job, given its simplicity, and it can actually do quite well for very small sample sizes, as compared to the other classification rules. The NNET performed well for 2 variables, but its performance quickly degrades as the number of genes increases, which can be explained by the high complexity of this classification rule, which leads to overfitting. CART and 1NN do not perform well with this data set, due to severe overfitting (even with the regularizing stopping criterion used for CART).

3.3. Regularization

Thus far we have taken the perspective that a collection of features is given, sample data are obtained, and a classifier based on the features is designed from the data via a classification rule. The feature set and sample data are taken as given, and the designer selects a classification rule. In this section, we consider alterations to this paradigm in order to deal with the small-sample problem, more specifically, a sample that is small relative to the number of features and classifier complexity. These methods fall under the general category of *regularization*.

3.3.1. Regularized discriminant analysis

The small-sample problem for QDA can be appreciated by considering the spectral decompositions of the covariance matrices,

$$\mathbf{K}_k = \sum_{j=1}^{d} \lambda_{kj} \mathbf{v}_{kj} \mathbf{v}_{kj}', \tag{3.29}$$

where $\lambda_{k1}, \lambda_{k2}, \ldots, \lambda_{kd}$ are the eigenvalues of $\mathbf{K}_k$ in decreasing order and $\mathbf{v}_{kj}$ is the eigenvector corresponding to λ_{kj}. Then it can be shown that the quadratic discriminant of (3.20) takes the form

$$d_k(\mathbf{x}) = -\sum_{j=1}^{d} \frac{\left[\mathbf{v}_{kj}(\mathbf{x} - \mathbf{u}_k)\right]^2}{\lambda_{kj}} - \sum_{j=1}^{d} \log \lambda_{kj} + 2 \log f_Y(k). \tag{3.30}$$

The discriminant is strongly influenced by the smallest eigenvalues. This creates a difficulty because the large eigenvalues of the sample covariance matrix are high-biased and the small eigenvalues are low-biased—and this phenomenon is accentuated for small samples.

Relative to QDA, a simple method of regularization is to apply LDA, even though the covariance matrices are not equal. This means estimating a single covariance matrix by pooling the data. This reduces the number of parameters to be estimated and increases the sample size relative to the smaller set of parameters. Generally, regularization reduces variance at the cost of bias, and the goal is substantial variance reduction with negligible bias.

A softer approach than strictly going from QDA to LDA is to shrink the individual covariance estimates in the direction of the pooled estimate. This can be accomplished by introducing a parameter α between 0 and 1 and using the estimates

$$\hat{\mathbf{K}}_k(\alpha) = \frac{n_k(1-\alpha)\hat{\mathbf{K}}_k + n\alpha\hat{\mathbf{K}}}{n_k(1-\alpha) + n\alpha}, \tag{3.31}$$

where n_k is the number of points corresponding to $Y = k$, $\hat{\mathbf{K}}_k$ is the sample covariance matrix for class k, and $\hat{\mathbf{K}}$ is the pooled estimate of the covariance matrix. QDA results from $\alpha = 0$ and LDA from $\alpha = 1$, with different amounts of shrinkage occurring for $0 < \alpha < 1$ [35]. While reducing variance, one must be prudent in choosing α, especially when the covariance matrices are very different.

To get more regularization while not overly increasing bias, one can shrink the regularized sample covariance matrix $\hat{\mathbf{K}}_k(\alpha)$ towards the identity multiplied by the average eigenvalue of $\hat{\mathbf{K}}_k(\alpha)$. This has the effect of decreasing large eigenvalues and increasing small eigenvalues, thereby offsetting the biasing effect seen in (3.30) [36]. Thus, we consider the estimate

$$\hat{\mathbf{K}}_k(\alpha,\beta) = (1-\beta)\hat{\mathbf{K}}_k(\alpha) + \frac{\beta}{n}\,\mathrm{tr}\,[\hat{\mathbf{K}}_k(\alpha)]\mathbf{I}, \tag{3.32}$$

where $\mathrm{tr}[\hat{\mathbf{K}}_k(\alpha)]$ is the trace of $\hat{\mathbf{K}}_k(\alpha)$, $\mathbf{I}$ is the identity, and $0 \le \beta \le 1$. To apply this *regularized discriminant analysis* using $\hat{\mathbf{K}}_k(\alpha,\beta)$ requires selecting two model parameters. Model selection is critical to advantageous regularization, and typically is problematic; nevertheless, simulation results for Gaussian conditional distributions indicate significant benefit of regularization for various covariance models, and very little increase in error, even in models where it does not appear to help.

3.3.2. Noise injection

Rather than regularizing the estimated covariance matrix, one can regularize the data itself by *noise injection*. This can be done by "spreading" the sample data, by means of synthetic data generated about each sample point, thereby creating a large synthetic sample from which to design the classifier while at the same time making the designed classifier less dependent on the specific points in the small data set. For instance, one may place a circular Gaussian distribution at each sample point, randomly generate points from each distribution, and then apply a classification rule. Such a Monte Carlo approach has been examined relative to LDA [37]. A spherical distribution need not be employed. Indeed, it has been demonstrated that it can be advantageous to base noise injection at a sample point based on the NNs of the point [37]. This kind of noise injection is not limited to any particular classification rule; however, it can be posed analytically in terms of matrix operations for linear classification, and this is critical to situations in which a large number of feature sets must be examined, in particular, microarray-based classification, where a vast number of potential feature sets are involved [9].

Noise injection can take a different form in which the sample data points themselves are perturbed by additive noise instead of new synthetic points being generated. This approach has been used in designing NNETs (of which linear classifiers are a special case), in particular, where owing to a small sample, the same data points are used repeatedly [38].

3.3.3. Error regularization

Rather than considering a single class from which to choose a classifier, one can consider a sequence of classes $\mathcal{C}_1, \mathcal{C}_2, \ldots$, find the best classifier in each class according to the data, and then choose among these according to which class is of appropriate complexity for the sample size. For instance, one might assume a nested sequence $\mathcal{C}_1 \subset \mathcal{C}_2 \subset \cdots$. The idea is to define a new measurement that takes into account both the error estimate of a designed classifier and the complexity of the class from which it has been chosen—the more complex the class, the larger the penalty. In this vein, we define a new *penalized error* that is a sum of the estimated error and a complexity penalty $\rho(n)$,

$$\tilde{\varepsilon}_n[\psi] = \hat{\varepsilon}_n[\psi] + \rho(n). \tag{3.33}$$

Relative to the constraint sequence $\{C^j\}$, *structural risk minimization* proceeds by selecting the classifier in each class that minimizes the empirical error over the sample, and then choosing among these the one possessing minimal penalized error, where in each case the penalty is relative to the class containing the classifier [39, 40].

Minimum-description-length (MDL) complexity regularization replaces error minimization by minimization of a sum of entropies, one relative to encoding the error and the other relative to encoding the classifier description, in an effort to balance increased error and increased model complexity [41, 42]. The MDL approach has been employed for microarray-based prediction [22, 43].

3.3.4. Feature selection

The feature-selection problem is to select a subset of k features from a set of n features that provides an optimal classifier with minimum error among all optimal classifiers for subsets of a given size. For instance, for the large number of expression measurements on a cDNA microarray, it is necessary to find a small subset with which to classify. The inherent combinatorial nature of the problem is readily seen from the fact that all k-element subsets must be checked to assure selection of the optimal k-element feature set [44].

An issue concerning error estimation is monotonicity of the error measure. The Bayes error is monotone: if A and B are feature sets for which $A \subset B$, then $\varepsilon_B \leq \varepsilon_A$, where ε_A and ε_B are the Bayes errors corresponding to A and B, respectively. However, if $\varepsilon_{A,n}$ and $\varepsilon_{B,n}$ are the corresponding errors resulting from

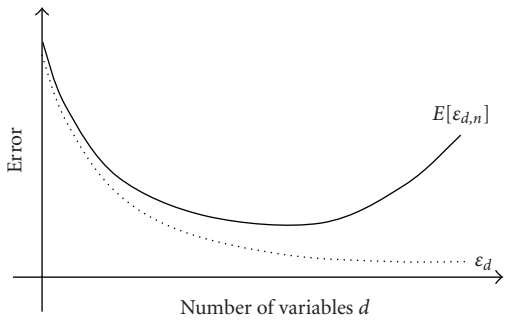


Figure 3.5. Bayes error and expected error versus number of features.

designed classifiers on a sample of size n, then it cannot be asserted that $E[\varepsilon_{B,n}]$ does not exceed $E[\varepsilon_{A,n}]$. Indeed, it is typical to observe a "peaking phenomenon" for fixed sample size, whereby the expected error decreases at first and then increases, for increasingly large feature sets. Thus, monotonicity does not apply for the expected error. This is illustrated in Figure 3.5, where the Bayes error ε_d and the expected error $E[\varepsilon_{d,n}]$ of the designed filter are plotted against the number of variables d. We can see that ε_d decreases, whereas $E[\varepsilon_{d,n}]$ decreases and then increases. We remark that the peaking phenomenon is referred to by some authors as the *Hughes phenomenon* [45, 46]. Note that, were $E[\varepsilon_{d,n}]$ known, then we could conclude that ε_d is no worse than $E[\varepsilon_{d,n}]$; however, we have only estimates of the error $\varepsilon_{d,n}$, which for small samples can be above or below ε_d.

A full exhaustive search can be mitigated by using a branch and bound feature-selection algorithm that takes advantage of the monotonicity property of the Bayes error [47]. If $A \subset B$ and C is a feature set for which $\varepsilon_C \geq \varepsilon_A$, then $\varepsilon_C \geq \varepsilon_B$. In principle, this approach yields an optimal solution; however, it suffers from two problems. First, worst-case performance can be exponentially complex, thereby making its use less attractive for very large feature sets; and second, estimation of the Bayes error must be used and therefore monotonicity is lost, a problem that is exacerbated by small samples. As is generally true with feature-selection methods, other criteria besides the Bayes error can be used to select features, monotonic criteria being necessary for strict application of the branch and bound algorithm. Even with the loss of monotonicity, the branch-and-bound approach may still provide good results [48].

When considering a large collection of features, the branch-and-bound technique is not sufficiently computationally efficient and suboptimal approaches need to be considered. The most obvious approach is to consider each feature by itself and choose the k features that perform individually the best. While easy, this method is subject to choosing a feature set with a large number of redundant features, thereby obtaining a feature set that is much worse than the optimal. Moreover, features that perform poorly individually may do well in combination with other features [9].

Perhaps the most common approach to suboptimal feature selection is *sequential selection*, either forward or backward, and their variants. Forward selection begins with a small set of features, perhaps one, and iteratively builds the feature set. Backward selection starts with a large set and iteratively reduces it. Owing to simpler calculations, forward selection is generally faster, and we will restrict our attention to it, noting that analogous comments apply to backwards selection. Here again, monotonicity issues and the "peaking" phenomenon arise: adjoining variables stepwise to the feature vector decreases the Bayes error but can increase errors of the designed filters.

Being more precise relative to forward selection, if A is the feature set at a particular stage of the algorithm and Q is the full set of potential features, then all sets of the form $A \cup \{X\}$ are considered, with $X \in Q - A$, and the feature X is adjoined to A if it is the one for which the error $\varepsilon[A \cup \{X\}]$ is minimal. An obvious problem is that once a feature is in the growing feature set, it cannot be removed. This problem can be handled by adjoining two or more features by considering sets of the form $A \cup B$, where B is a subset of $Q - A$ possessing b features, and then deleting features by considering sets of the form $A \cup B_0 - C$, where B_0 has been chosen on the adjoining part of the iteration and C is a subset of $A \cup B_0$ possessing $c < b$ features. At each stage of the iteration, the feature set is grown by $b - c$ features. While growing by adjoin-delete iterations is superior to just adjoining, there is still inflexibility owing to the a priori choices of b and c. Flexibility can be added to sequential forward selection by considering *sequential forward floating selection* (*SFFS*), where the number of features to be adjoined and deleted is not fixed, but is allowed to "float" [49].

When there is a large number of potential random variables for classification, feature selection is problematic and the best method to use depends on the circumstances. Evaluation of methods is generally comparative and based on simulations [50].

3.3.5. Feature extraction

Rather than reducing dimensionality by selecting from the original features, one might take the approach of *feature extraction*, where a transform is applied to the original features to map them into a lower dimensional space. Since the new features involve a transform of the original features, the original features remain (although some may be eliminated by compression) and are still part of the classification process. A disadvantage of feature extraction is that the new features lack the physical meaning of the original features—for instance, gene expression levels. A potential advantage of feature extraction is that, given the same number of reduced features, the transform features may provide better classification than selected individual features. Perhaps the most common form of feature extraction is *principal component analysis* (PCA).

Consider the (random) observation vector $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, where the observations have been normalized to have zero means. Since the covariance matrix $\mathbf{K}$ is symmetric, if λ_1 and λ_2 are distinct eigenvalues, then their respective

eigenvectors will be orthogonal and the desired orthonormal eigenvectors can be found by dividing each by its own magnitude. On the other hand, if an eigenvalue has repeated eigenvectors, then these will be linearly independent and an algebraically equivalent set can be found by the Gram-Schmidt orthogonalization procedure.

According to the Karhunen-Loeve theorem, if the vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ are the orthonormalized eigenvectors of $\mathbf{K}$ corresponding to the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, then

$$\mathbf{X} = \sum_{i=1}^{n} Z_i \mathbf{u}_i, \tag{3.34}$$

where $Z_1, Z_2, \ldots, Z_n$ are uncorrelated and given by $Z_i = \mathbf{X} \cdot \tilde{\mathbf{u}}_i$. The values $Z_1, Z_2, \ldots, Z_n$ are called the *principal components* for $\mathbf{X}$. For $m < n$, data compression is achieved by approximating $\mathbf{X}$ by

$$\mathbf{X}_m = \sum_{i=1}^{m} Z_i \mathbf{u}_i. \tag{3.35}$$

The *mean-square error* between $\mathbf{X}$ and $\mathbf{X}_m$ is given by

$$E[\mathbf{X}, \mathbf{X}_m] = \sum_{k=1}^{n} E\Big[|X_k - X_{m,k}|^2 \Big], \tag{3.36}$$

where the components of $\mathbf{X}_m$ are $X_{m,1}, X_{m,2}, \ldots, X_{m,n}$. It can be shown that

$$E[\mathbf{X}, \mathbf{X}_m] = \sum_{k=m+1}^{n} \lambda_k. \tag{3.37}$$

Since the eigenvalues are decreasing with increasing k, the error is minimized by keeping the first m terms. To apply PCA for the purposes of feature extraction, $Z_1, Z_2, \ldots, Z_m$ are employed.

3.4. Error estimation

Error estimation is a key aspect of classification, as it impacts both classifier design and variable selection. Recall that the performance measure of a designed classifier is the "true" error ε_n, whereas the performance measure of a classification rule (for fixed sample size n) is the expected error $E[\varepsilon_n]$. However, both these quantities can only be computed exactly if one knows the feature-label distribution for the classification problem. Since in practice such knowledge is rarely, if ever, at hand, one needs to estimate the true error from the available sample data.

An error estimator $\hat{\varepsilon}_n$ may be a deterministic function of the sample data S_n, in which case it is a *nonrandomized error estimator*. Such an error estimator is random

only through the random sample. Among popular nonrandomized error estimators, we have resubstitution, leave-one-out (LOO), and fixed-fold cross-validation. By contrast, *randomized error estimators* have "internal" random factors that affect their outcome. Popular randomized error estimators include random-fold cross-validation and all bootstrap error estimators (all aforementioned error estimators will be discussed in detail below).

A key feature that often dictates the performance of an error estimator is its variance, especially in small-sample settings. The *internal variance* of an error estimator is the variance due only to its internal random factors, $V_{\text{int}} = \text{Var}(\hat{\varepsilon}_n | S_n)$. This variance is zero for nonrandomized error estimators. The full variance $\text{Var}(\hat{\varepsilon}_n)$ of the error estimator is the one we are really concerned about, since it takes into account the uncertainty introduced by the random sample data. Using the well-known conditional-variance formula, $\text{Var}(X) = E[\text{Var}(X|Y)] + \text{Var}(E[X|Y])$ [51], one can break down $\text{Var}(\hat{\varepsilon}_n)$ as

$$\text{Var}\left(\hat{\varepsilon}_n\right) = E\left[V_{\text{int}}\right] + \text{Var}\left(E\left[\hat{\varepsilon}_n | S_n\right]\right). \tag{3.38}$$

The second term on the right-hand side is the one that includes the variability due to the random sample. Note that, for nonrandomized $\hat{\varepsilon}_n$, we have $V_{\text{int}} = 0$ and $E[\hat{\varepsilon}_n | S_n] = \hat{\varepsilon}_n$. For randomized error estimators, the first term on the right-hand side has to be made small through intensive computation, in order to achieve small overall estimator variance. This is one of the reasons why randomized error estimators are typically very inefficient computationally, as we will see below.

3.4.1. Holdout estimation

We now proceed to discuss specific error-estimation techniques. If there is an abundance of sample data, then they can be split into *training data* and *test data*. A classifier is designed on the training data, and its estimated error is the proportion of errors it makes on the test data. We denote this test-data error estimate by $\hat{\varepsilon}_{n,m}$, where m is the number of sample pairs in the test data. Since the test data are random and independent from the training data, this is a randomized error estimator. It is unbiased in the sense that, given the training data S_n, $E[\hat{\varepsilon}_{n,m} | S_n] = \varepsilon_n$, and thus $E[\hat{\varepsilon}_{n,m}] = E[\varepsilon_n]$. The internal variance of this estimator can be bounded as follows [18]:

$$V_{\text{int}} = E\left[\left(\hat{\varepsilon}_{n,m} - \varepsilon_n\right)^2 \middle| S_n\right] \leq \frac{1}{4m} \tag{3.39}$$

which tends to zero as $m \to \infty$. Moreover, by using (3.38), we get that the full variance of the holdout estimator is simply

$$\text{Var}\left(\hat{\varepsilon}_{n,m}\right) = E\left[V_{\text{int}}\right] + \text{Var}\left[\varepsilon_n\right]. \tag{3.40}$$

Thus, provided that m is large, so that V_{int} is small (this is guaranteed by (3.39) for large enough m), the variance of the holdout estimator is approximately equal to

the variance of the true error itself, which is typically small, provided n is not too small.

The problem with using both training and test data is that, in practice, one often does not usually have available a large enough data set to be able to make both n and m large enough. For example, in order to get the standard-deviation bound in (3.39) down to an acceptable level, say 0.05, it would be necessary to use 100 test samples. On the other hand, data sets that contain fewer than 100 overall samples are quite common. Therefore, for a large class of practical problems, where samples are at a premium, holdout error estimation is effectively ruled out. In such cases, one must use the same data for training and testing.

3.4.2. Resubstitution

One approach is to use all sample data to design a classifier ψ_n, and estimate ε_n by applying ψ_n to the same data. The *resubstitution* estimate $\hat{\varepsilon}_n^R$ is the fraction of errors made by ψ_n on the training data:

$$\hat{\varepsilon}_n^R = \frac{1}{n}\sum_{i=1}^{n} \left| Y_i - \psi_n(\mathbf{X}_i) \right|. \tag{3.41}$$

This is none other than the apparent error rate mentioned in Section 3.2.3. Resubstitution is usually low-biased, meaning $E[\hat{\varepsilon}_n^R] \leq E[\varepsilon_n]$—but not always. For fixed-partition histogram rules, meaning those that are independent of the sample size and the data, the resubstitution error estimate is low-biased and its variance is bounded in terms of the sample size by $\mathrm{Var}[\hat{\varepsilon}_n^R] \leq 1/n$. For small samples, the bias can be severe. It typically improves for large samples. Indeed, for fixed-partition histogram rules, $E[\hat{\varepsilon}_n^R]$ is monotonically increasing. The mean-square error for resubstitution error estimation for a fixed-partition histogram rule having at most q cells possesses the bound [18]

$$E\left[\left| \hat{\varepsilon}_n^R - \varepsilon_n \right|^2 \right] \leq \frac{6q}{n}. \tag{3.42}$$

3.4.3. Cross-validation

With cross-validation, classifiers are designed from parts of the sample, each is tested on the remaining data, and ε_n is estimated by averaging the errors. In *k-fold cross-validation*, S_n is partitioned into k folds $S_{(i)}$, for $i = 1, 2, \ldots, k$, where for simplicity, we assume that k divides n. Each fold is left out of the design process and used as a test set, and the estimate is the overall proportion of errors committed on all folds:

$$\hat{\varepsilon}_{n,k}^{CV} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n/k} \left| Y_j^{(i)} - \psi_{n,i}(\mathbf{X}_j^{(i)}) \right|, \tag{3.43}$$

where $\psi_{n,i}$ is designed on $S_n - S_{(i)}$ and $(\mathbf{X}_j^{(i)}, Y_j^{(i)})$ is a sample point in $S_{(i)}$. Picking the folds randomly leads to *random-fold cross-validation*. On the other hand, preselecting which parts of the sample to go into each fold leads to *fixed-fold cross-validation*, a nonrandomized error estimator. The process may be repeated, where several cross-validated estimates are computed, using different partitions of the data into folds, and the results are averaged. In *stratified cross-validation*, the classes are represented in each fold in the same proportion as in the original data. A k-fold cross-validation estimator is unbiased as an estimator of $E[\varepsilon_{n-n/k}]$, that is, $E[\hat{\varepsilon}_{n,k}^{CV}] = E[\varepsilon_{n-n/k}]$.

A *leave-one-out* estimator is an n-fold cross-validated estimator. A single observation is left out, n classifiers are designed from sample subsets formed by leaving out one sample pair, each is applied to the left-out pair, and the estimator $\hat{\varepsilon}_n^{CV}$ is $1/n$ times the number of errors made by the n classifiers (where for notational ease, we write $\hat{\varepsilon}_n^{CV}$ instead of $\hat{\varepsilon}_{n,n}^{CV}$). Note that both fixed-$n$-fold and random-$n$-fold cross-validated estimators coincide with the same nonrandomized LOO estimator. The estimator $\hat{\varepsilon}_n^{CV}$ is unbiased as an estimator of $E[\varepsilon_{n-1}]$, that is, $E[\hat{\varepsilon}_n^{CV}] = E[\varepsilon_{n-1}]$. A key concern is the variance of the estimator for small n [13]. Performance depends on the classification rule. The mean-square error for LOO error estimation for a fixed-partition histogram rule possesses the bound [18]

$$E\left[\left|\hat{\varepsilon}_n^{CV} - \varepsilon_n\right|^2\right] \leq \frac{1 + 6e^{-1}}{n} + \frac{6}{\sqrt{\pi(n-1)}}. \tag{3.44}$$

Comparing (3.42) and (3.44), we can see that $\sqrt{n-1}$ for LOO estimation as opposed to n in the denominator for resubstitution shows greater variance for LOO, for fixed-partition histogram rules.

To appreciate the difficulties inherent in the LOO bounds, we will simplify them in a way that makes them more favorable to precise estimation. The performance of $\hat{\varepsilon}_n^{CV}$, guaranteed by (3.44), becomes better if we lower the bound. A lower bound than the one in (3.44) is $1.8/\sqrt{n-1}$. Even for this better standard-deviation bound, the numbers one gets for $n = 50$ and 100 still exceed 0.5 and 0.435, respectively. So the bound is essentially useless for small samples.

3.4.4. Bootstrap

The bootstrap methodology is a general resampling strategy that can be applied to error estimation [52]. It is based on the notion of an *empirical distribution* F^*, which serves as a replacement to the original unknown feature-label distribution F. The empirical distribution is discrete, putting mass $1/n$ on each of the n available data points. A *bootstrap sample* S_n^* from F^* consists of n equally likely draws with replacement from the original sample S_n. Hence, some points may appear multiple times, whereas others may not appear at all. The probability that a given point will not appear in S_n^* is $(1 - 1/n)^n \approx e^{-1}$. Therefore, a bootstrap sample of size n contains on average $(1 - e^{-1})n \approx (0.632)n$ of the original points. The basic

bootstrap error estimator is the *bootstrap zero estimator* [53], which is defined by

$$\hat{\varepsilon}_n^{BZ} = E_{F^*}[\,|\,Y - \psi_n^*(\mathbf{X})\,| : (\mathbf{X}, Y) \in S_n - S_n^*\,], \tag{3.45}$$

where S_n is fixed. The classifier ψ_n^* is designed on the bootstrap sample and tested on the points that are left out. In practice, the expectation is approximated by a sample mean based on B independent replicates $S_{n,b}^*$, $b = 1, 2, \ldots, B$:

$$\hat{\varepsilon}_n^{BZ} = \frac{\sum_{b=1}^{B}\sum_{i=1}^{n} |\,Y_i - \psi_{n,b}^*(\mathbf{X}_i)\,|\, I_{(\mathbf{X}_i, Y_i)\in S_n - S_{n,b}^*}}{\sum_{b=1}^{B}\sum_{i=1}^{n} I_{(\mathbf{X}_i, Y_i)\in S_n - S_{n,b}^*}}. \tag{3.46}$$

The bootstrap zero estimator is clearly a randomized error estimator. In order to keep the internal variance low, and thus achieve a small overall variance, a large enough number B of bootstrap samples must be employed. In the literature, B between 25 and 200 has been recommended. In addition, a variance-reducing technique is often employed, called balanced bootstrap resampling [54], according to which each sample point is made to appear exactly B times in the computation.

The bootstrap zero estimator tends to be a high-biased estimator of $E[\varepsilon_n]$, since the number of points available for design is on average only $0.632n$. The 0.632 *bootstrap estimator* tries to correct this bias by doing a weighted average of the zero and resubstitution estimators [53]:

$$\hat{\varepsilon}_n^{B632} = (1 - 0.632)\hat{\varepsilon}_n^R + 0.632\hat{\varepsilon}_n^{BZ}. \tag{3.47}$$

On the other hand, the *bias-corrected bootstrap* estimator tries to correct for resubstitution bias. It is defined by

$$\hat{\varepsilon}_n^{BBC} = \hat{\varepsilon}_n^R + \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{n}\left(\frac{1}{n} - P_{i,b}^*\right)|\,Y_i - \psi_{n,b}^*(\mathbf{X}_i)\,|, \tag{3.48}$$

where $P_{i,b}^*$ is the proportion of times that $(\mathbf{X}_i, Y_i)$ appears in the bootstrap sample $S_{n,b}^*$. This estimator adds to the resubstitution estimator the bootstrap estimate of its bias.

3.4.5. Bolstering

A quick calculation reveals that the resubstitution estimator is given by

$$\hat{\varepsilon}_n^R = E_{F^*}[\,|\,Y - \psi_n(\mathbf{X})\,|\,], \tag{3.49}$$

where F^* is the empirical feature-label distribution. Relative to F^*, no distinction is made between points near or far from the decision boundary. If one spreads the probability mass at each point of the empirical distribution, then variation is reduced because points near the decision boundary will have more mass on the other

side than will points far from the decision boundary. Hence, more confidence is attributed to points far from the decision boundary than to points near it.

To take advantage of this observation, consider a probability density function $f_i^\diamond$, for $i = 1, 2, \ldots, n$, called a *bolstering kernel*, and define the *bolstered empirical distribution* $F^\diamond$, with probability density function given by

$$f^\diamond(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i^\diamond(\mathbf{x} - \mathbf{x}_i). \tag{3.50}$$

The *bolstered resubstitution* estimator [55] is obtained by replacing F^* by $F^\diamond$ in (3.49):

$$\hat{\varepsilon}_n^{\diamond R} = E_{F^\diamond}[\,|\, Y - \psi_n(\mathbf{X}) \,|\,]. \tag{3.51}$$

Bolstering may actually be applied to any error-counting estimation procedure; for example, one can define a *bolstered leave-one-out (BLOO)* estimator [55]. However, in what follows, we focus, for the most part, on the bolstered resubstitution case.

The following is a computational expression, equivalent to (3.51), for the bolstered resubstitution estimator:

$$\hat{\varepsilon}_n^{\diamond R} = \frac{1}{n}\sum_{i=1}^{n}\left(I_{y_i=0}\int_{A_1} f_i^\diamond(x - x_i)\,dx + I_{y_i=1}\int_{A_0} f_i^\diamond(x - x_i)\,dx\right). \tag{3.52}$$

The integrals are the error contributions made by the data points, according to whether $y_i = 0$ or $y_i = 1$. The bolstered resubstitution estimate is equal to the sum of all error contributions divided by the number of points. If the classifier is linear, then the decision boundary is a hyperplane and it is usually possible to find analytical expressions for the integrals, for instance, for Gaussian bolstering; otherwise, Monte Carlo integration can be employed, and experience shows that very few Monte Carlo samples are necessary. See Figure 3.6 for an illustration, where the classifier is linear and the bolstering kernels are uniform circular distributions (note that the bolstering kernels need not have the same variance). The samples in this example correspond to a subset of the cancer data used in Section 3.2.5 (the linear classifier in Figure 3.6 was obtained via LDA).

A key issue is choosing the amount of bolstering, that is, the kernel variances. Since the purpose of bolstering is to improve error estimation in the small-sample setting, we do not want to use bolstering kernels that require complicated inferences. Hence, we consider zero-mean, spherical bolstering kernels with covariance matrices of the form $\sigma_i \mathbf{I}$. The choice of the parameters $\sigma_1, \sigma_2, \ldots, \sigma_n$ determines the variance and bias properties of the corresponding bolstered estimator. If $\sigma_1 = \sigma_2 = \cdots = \sigma_n = 0$, then there is no bolstering and the bolstered estimator reduces to the original estimator. As a general rule, larger σ_i lead to lower-variance estimators, but after a certain point, this advantage is offset by increasing bias.

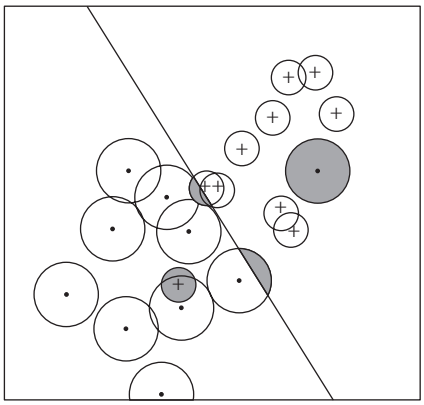


Figure 3.6. Bolstered resubstitution for linear classification, assuming uniform circular bolstering kernels. The bolstered resubstitution error is the sum of all contributions (shaded areas) divided by the number of points.

We wish to select $\sigma_1, \sigma_2, \ldots, \sigma_n$ to make the bolstered resubstitution estimator nearly unbiased. One can think of $(\mathbf{X}, Y)$ in (3.1) as a random test point. Given $Y = k$, this test point is at a "mean distance" δ_k from the data points belonging to class k, for $k = 1, 2$. Resubstitution tends to be optimistically biased because the test points in (3.41) are all at distance zero from the training data. Since bolstered resubstitution spreads the test points, the task is to find the amount of spreading that makes the test points to be as close as possible to the true mean distance to the training data points.

The mean distance δ_k can be approximated by the sample-based estimate

$$\hat{\delta}_k = \frac{\sum_{i=1}^{n} \min_{j \neq i} \{||\mathbf{x}_i - \mathbf{x}_j||\} I_{y_i=k}}{\sum_{i=1}^{n} I_{y_i=k}}. \tag{3.53}$$

This estimate is the mean minimum distance between points belonging to class $Y = k$.

Rather than estimating a separate bolstering kernel standard deviation for each sample point, we propose to compute two distinct standard deviations τ_1 and τ_2, one for each class, based on the mean-distance estimates δ_1 and δ_2 (this limits the complexity of the estimation problem, which is advantageous in small-sample settings). Thus, we let $\sigma_i = \tau_k$, for $y_i = k$. To arrive at estimates for τ_1 and τ_2, let D be the random variable giving the distance to the origin of a randomly selected point from a unit-variance bolstering kernel, and let F_D be the probability distribution function for D. In the case of a bolstering kernel of standard deviation τ_k, all distances get multiplied by τ_k, so if D' is the distance random variable for this more general case, then $F_{D'}(x) = F_D(x/\tau_k)$. For the class $Y = k$, the value of τ_k is to be chosen so that the median distance of a random test point to the origin of the bolstering kernel of standard deviation τ_k is equal to the mean-distance $\hat{\delta}_k$, the result being that half of the test points will be farther from the origin than $\hat{\delta}_k$,

and the other half will be closer. A little reflection shows that τ_k is the solution of the equation $F_{D'}^{-1}(0.5) = \tau_k F_D^{-1}(0.5) = \widehat{\delta}_k$, so that the final estimated standard deviations for the bolstering kernels are given by

$$\sigma_i = \frac{\widehat{\delta}_k}{F_D^{-1}(0.5)} \quad \text{for } y_i = k. \tag{3.54}$$

Note that, as the number of samples in the sample increases, $\widehat{\delta}_k$ decreases, and therefore so does σ_i. This is the expected behavior in this case, since plain resubstitution tends to be less biased as the number of samples increases. Note also that the denominator $F_D^{-1}(0.5)$ may be viewed as a constant dimensionality correction factor (being a function of the number of dimensions through D), which can be precomputed and stored offline.

We mention that, when resubstitution is heavily low-biased, it may not be a good idea to spread incorrectly classified data points, as that increases optimism of the error estimator (low bias). Bias is reduced by letting $\sigma_i = 0$ (no bolstering) for incorrectly classified points. The resulting estimator is called the *semi-bolstered resubstitution* estimator [55].

3.4.6. Error-estimation performance

We now illustrate the small-sample performance of several of the error estimators discussed in the previous subsections by means of simulation experiments based on synthetic data (see also [13, 55]). We consider resubstitution (resub), leave-one-out (LOO), stratified 10-fold cross-validation with 10 repetitions (CV10r), the balanced 0.632 bootstrap (b632), Gaussian bolstered resubstitution (bresub), Gaussian semibolstered resubstitution (sresub) and Gaussian BLOO. The number of bootstrap samples is $B = 100$, which makes the number of designed classifiers be the same as for CV10r. We employ three classification rules; in order of complexity, LDA, 3NN, and CART. For LDA, the bolstered estimators are computed using analytical formulas developed in [55]; for 3NN and CART, Monte Carlo sampling is used—we have found that only $M = 10$ Monte Carlo samples per bolstering kernel are adequate, and increasing M to a larger value reduces the variance of the estimators only slightly.

The experiments assess the empirical distribution of $\varepsilon_n - \widehat{\varepsilon}_n$ for each error estimator $\widehat{\varepsilon}_n$. This distribution measures the difference between the true error and the estimated error of the designed classifier. Deviation distributions are from 1000 independent data sets drawn from several models. The synthetic model for LDA consists of spherical Gaussian class-conditional densities with means located at $(\delta, \delta, \ldots, \delta)$ and $(-\delta, -\delta, \ldots, -\delta)$, where $\delta > 0$ is a separation parameter that controls the Bayes error. The synthetic model for 3NN and CART corresponds to class-conditional densities given by a mixture of spherical Gaussians, with means at opposing vertices of a hypercube centered at the origin and side 2δ. For instance, for $d = 5$, the class-conditional density for class 0 has means at $(\delta, \delta, \delta, \delta, \delta)$ and $(-\delta, -\delta, -\delta, \delta, -\delta)$, and the class-conditional density for class 1 has means at

Table 3.1. Twelve experiments used in the simulation study.

Experiment	Rule	d	δ	σ_1	σ_2	Bayes error
1	LDA	2	0.59	1.00	1.00	0.202
2	LDA	2	0.59	1.00	4.00	0.103
3	LDA	5	0.37	1.00	1.00	0.204
4	LDA	5	0.37	1.00	2.16	0.103
5	3NN	2	1.20	1.00	1.00	0.204
6	3NN	2	1.20	1.00	5.20	0.103
7	3NN	5	0.77	1.00	1.00	0.204
8	3NN	5	0.77	1.00	2.35	0.105
9	CART	2	1.20	1.00	1.00	0.204
10	CART	2	1.20	1.00	5.20	0.103
11	CART	5	0.77	1.00	1.00	0.204
12	CART	5	0.77	1.00	2.35	0.105

$(\delta, -\delta, \delta, -\delta, \delta)$ and $(-\delta, \delta, -\delta, \delta, -\delta)$. In all cases, there are equal a priori class probabilities.

Table 3.1 lists the parameters for the twelve experiments considered in this study, corresponding to choices among the three classification rules, for various separations δ between the class means, low or moderate dimensionality d, and equal or distinct standard deviations σ_1 and σ_2 for the class-conditional densities. The parameters were chosen so as to give Bayes error of about 0.1 in half of the cases and about 0.2 in the other half. These are difficult models, with considerable overlapping between the classes (even in the cases where the Bayes error is 0.1) owing to large discrepancy in variance between the classes, not to actual separation between the means.

Plots of beta-density fits of the empirical deviation distribution for sample size $n = 20$ are displayed in Figures 3.7, 3.8, and 3.9 (see [55] and its companion website for the complete results of this simulation study). Note that the narrower and taller the distribution, the smaller the variance of the deviation, whereas the closer its mean is to the vertical axis, the more unbiased the error estimator is. We can see that resubstitution, LOO, and even 10-fold cross-validation are generally outperformed by the bootstrap and bolstered estimators. Bolstered resubstitution is very competitive with the bootstrap, in some cases outperforming it. For LDA, the best estimator overall is bolstered resubstitution. For 3NN and CART, which are classifiers known to overfit in small-sample settings, the situation is not so clear. For 3NN, we can see that bolstered resubstitution fails in correcting the bias of resubstitution for $d = 5$, despite having small variance (note that it is still the best overall estimator for 3NN in Experiment 5). For CART, the bootstrap estimator is affected by the extreme low-biasedness of resubstitution. In this case, bolstered resubstitution performs much better than in the 3NN case, but the best overall estimator is the semibolstered resubstitution. The BLOO is generally much more variable than the bolstered resubstitution estimators, but it displays much less bias.

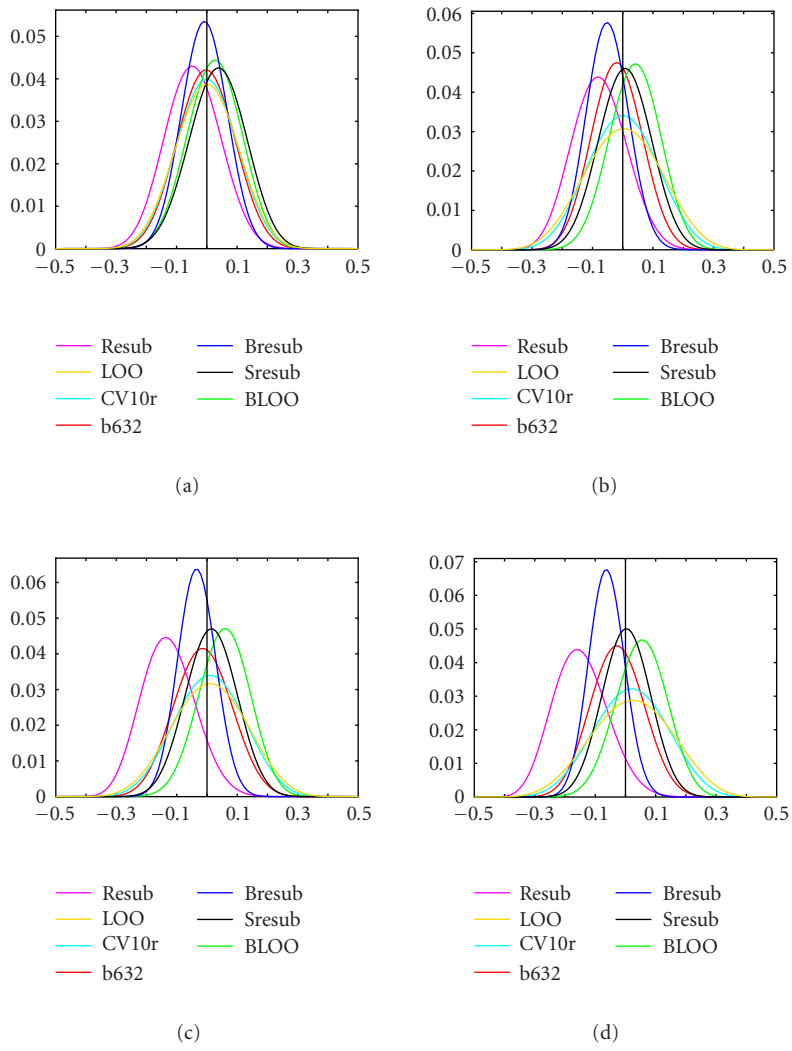


Figure 3.7. Beta fits to empirical deviation distribution for LDA and $n = 20$. (a) Experiment 1, (b) experiment 2, (c) experiment 3, and (d) experiment 4.

Computation time can be critical. We have found that resubstitution is by far the fastest estimator, with its bolstered versions coming just behind. LOO and its bolstered version are fast for a small number of samples, but performance quickly degrades with an increasing number of samples. The 10-fold cross-validation and the bootstrap estimator are the slowest estimators. Bolstered resubstitution can be hundreds of times faster than the bootstrap estimator (see [55] for a listing of timings).

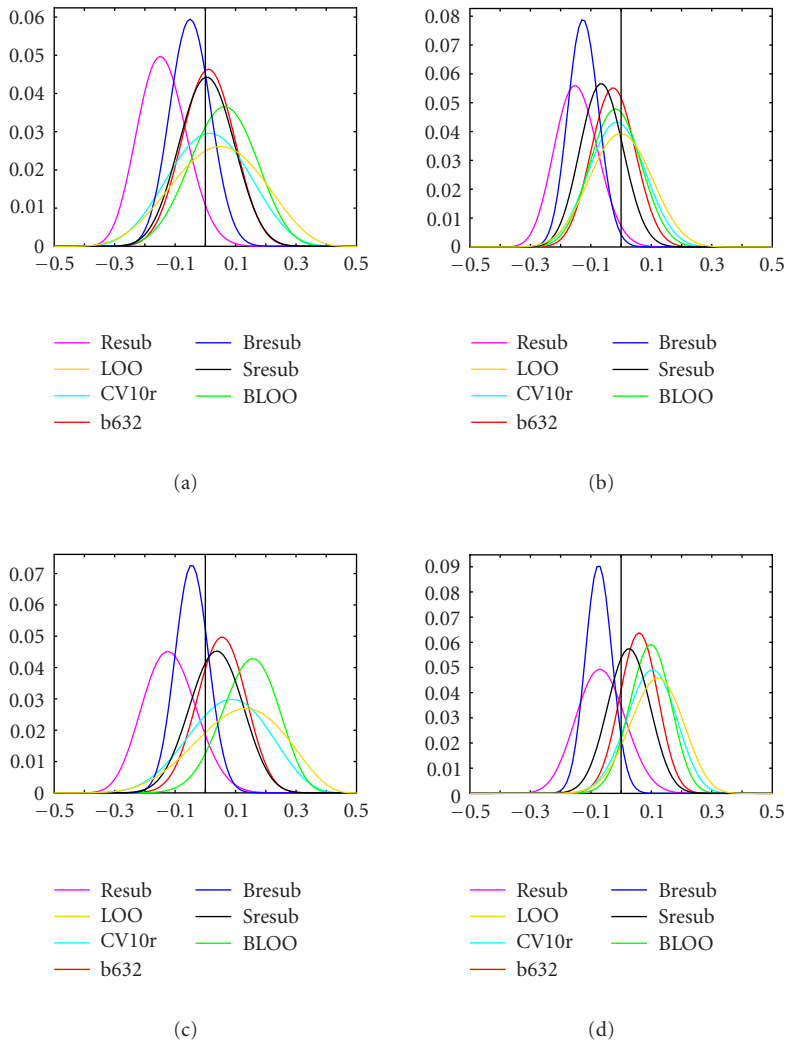


Figure 3.8. Beta fits to empirical deviation distribution for 3NN and $n = 20$. (a) Experiment 5, (b) experiment 8, (c) experiment 7, and (d) experiment 9.

We close this subsection with some comments on error estimation and the measurement of feature-set performance. Given a large number of potential feature sets, one may wish to rank them according to the performances of their optimal classifiers, which in practice means the performances of their designed classifiers. A recent study has addressed the impact of error estimation on feature selection [56]. The experiments indicate that the considered feature-selection algorithms can perform close to optimal (full search with true error) when the true

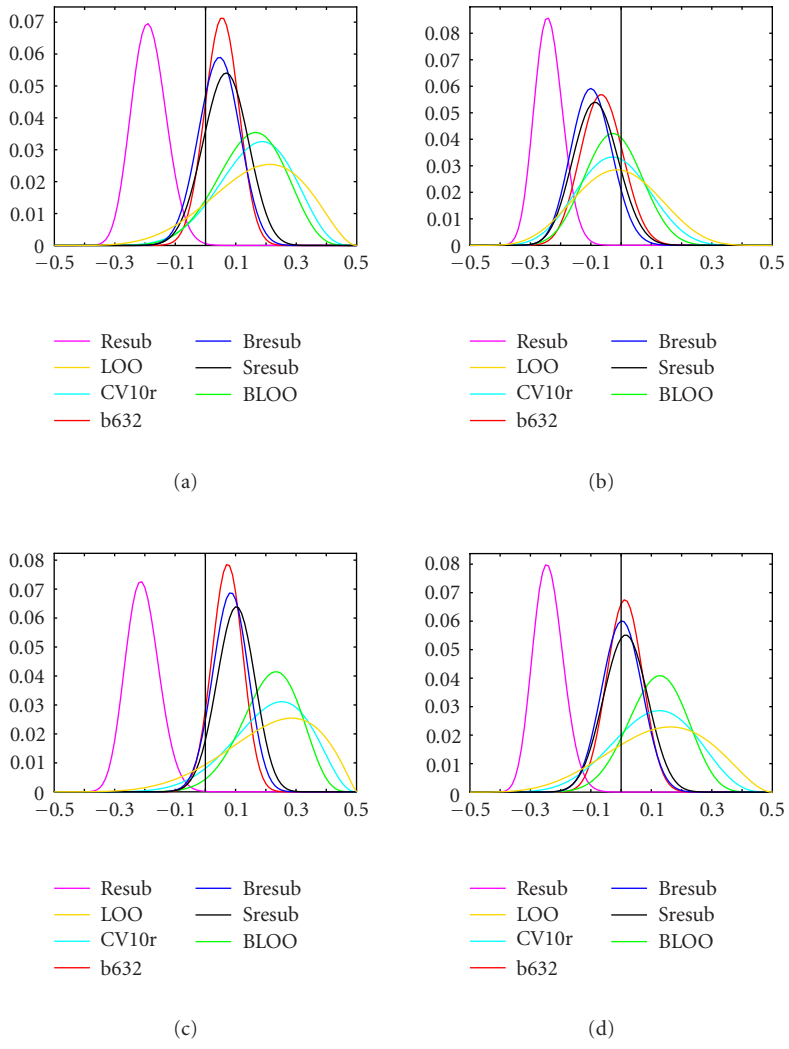


Figure 3.9. Beta fits to empirical deviation distribution for CART and $n = 20$. (a) Experiment 9, (b) experiment 10, (c) experiment 11, and (d) experiment 12.

error is employed in feature selection. With large samples, many error-estimation procedures work quite well so that one has good estimates of the true error; however, this is not the case with small samples. The study uses two performance measures for feature selection: comparison of the true error of the optimal feature set with the true error of the feature set found by a feature-selection algorithm, and the number of features among the truly optimal feature set that appear

in the feature set found by the algorithm. The study considers seven error estimators applied to three standard suboptimal feature-selection algorithms and exhaustive search, and it considers three different feature-label model distributions. It draws two conclusions for the cases considered: (1) depending on the sample size and the classification rule, feature-selection algorithms can produce feature sets whose corresponding classifiers possess errors far in excess of the classifier corresponding to the optimal feature set; and (2) for small samples, differences in performances among the feature-selection algorithms appear to be less significant than performance differences among the error estimators used to implement the algorithms. Moreover, keeping in mind that results depend on the particular classifier-distribution pair, for the error estimators used in the study, bootstrap and bolstered resubstitution usually outperform cross-validation, and bolstered resubstitution usually performs as well as or better than bootstrap.

Bibliography

[1] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 559–583, 2000.

[2] M. Bittner, P. Meltzer, Y. Chen, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.

[3] G. Callagy, E. Cattaneo, Y. Daigo, et al., "Molecular classification of breast carcinomas using tissue microarrays," *Diagn. Mol. Pathol.*, vol. 12, no. 1, pp. 27–34, 2003.

[4] L. Dyrskjot, T. Thykjaer, M. Kruhoffer, et al., "Identifying distinct classes of bladder carcinoma using microarrays," *Nat. Genet.*, vol. 33, no. 1, pp. 90–96, 2003.

[5] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[6] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[7] I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, et al., "Gene-expression profiles in hereditary breast cancer," *N. Engl. J. Med.*, vol. 344, no. 8, pp. 539–548, 2001.

[8] J. Khan, J. S. Wei, M. Ringner, et al., "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nat. Med.*, vol. 7, no. 6, pp. 673–679, 2001.

[9] S. Kim, E. R. Dougherty, I. Shmulevich, et al., "Identification of combination gene sets for glioma classification," *Mol. Cancer. Ther.*, vol. 1, no. 13, pp. 1229–1236, 2002.

[10] T. Kobayashi, M. Yamaguchi, S. Kim, et al., "Microarray reveals differences in both tumors and vascular specific gene expression in de novo cd5+ and cd5- diffuse large b-cell lymphomas," *Cancer Res.*, vol. 63, no. 1, pp. 60–66, 2003.

[11] A. S. Levenson, I. L. Kliakhandler, K. M. Svoboda, et al., "Molecular classification of selective oestrogen receptor modulators on the basis of gene expression profiles of breast cancer cells expressing oestrogen receptor alpha," *Br. J. Cancer*, vol. 87, no. 4, pp. 449–456, 2002.

[12] A. Szabo, K. Boucher, W. L. Carroll, L. B. Klebanov, A. D. Tsodikov, and A. Y. Yakovlev, "Variable selection and pattern recognition with gene expression data generated by the microarray technology," *Math. Biosci.*, vol. 176, no. 1, pp. 71–98, 2002.

[13] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?," *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[14] E. R. Dougherty, "Small sample issues for microarray-based classification," *Comparative and Functional Genomics*, vol. 2, no. 1, pp. 28–34, 2001.

[15] M. J. van de Vijver, Y. D. He, L. J. van't Veer, et al., "A gene-expression signature as a predictor of survival in breast cancer," *N. Engl. J. Med.*, vol. 347, no. 25, pp. 1999–2009, 2002.

[16] V. N. Vapnik and A. Ya. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory Probab. Appl.*, vol. 16, no. 2, pp. 264–280, 1971.

[17] E. B. Baum, "On the capabilities of multilayer perceptrons," *J. Complexity*, vol. 4, no. 3, pp. 193–215, 1988.

[18] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31 of *Applications of Mathematics (New York)*, Springer-Verlag, New York, NY, USA, 1996.

[19] L. Gordon and R. A. Olshen, "Asymptotically efficient solutions to the classification problem," *Ann. Statist.*, vol. 6, no. 3, pp. 515–533, 1978.

[20] E. R. Dougherty, M. Bittner, Y. Chen, et al., "Nonlinear filters in genomic control," in *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Antalya, Turkey, June 1999.

[21] S. Kim, E. R. Dougherty, M. L. Bittner, et al., "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *J. Biomed. Opt.*, vol. 5, no. 4, pp. 411–424, 2000.

[22] I. Tabus and J. Astola, "On the use of MDL principle in gene expression prediction," *EURASIP J. Appl. Signal Process.*, vol. 2001, no. 4, pp. 297–303, 2001.

[23] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, NY, USA, 1993.

[24] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From boolean to probabilistic boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[25] X. Zhou, X. Wang, and E. R. Dougherty, "Binarization of microarray data on the basis of a mixture model," *Mol. Cancer Ther.*, vol. 2, no. 7, pp. 679–684, 2003.

[26] I. Shmulevich and W. Zhang, "Binary analysis and optimization-based normalization of gene expression data," *Bioinformatics*, vol. 18, no. 4, pp. 555–565, 2002.

[27] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[28] C. J. Stone, "Consistent nonparametric regression. With discussion and a reply by the author," *Ann. Statist.*, vol. 5, no. 4, pp. 595–645, 1977.

[29] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[30] P. W. Wahl and R. A. Kronmal, "Discriminant functions when covariances are unequal and sample sizes are moderate," *Biometrics*, vol. 33, pp. 479–484, 1977.

[31] A. Farago and G. Lugosi, "Strong universal consistency of neural network classifiers," *IEEE Trans. Inform. Theory*, vol. 39, no. 4, pp. 1146–1151, 1993.

[32] C. M. Bishop, *Neural Networks for Pattern Recognition*, The Clarendon Press, Oxford University Press, New York, NY, USA, 1995.

[33] L. J. van't Veer, H. Dai, M. J. van de Vijver, et al., "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.

[34] A. J. Izenman, "Recent developments in nonparametric density estimation," *J. Amer. Statist. Assoc.*, vol. 86, no. 413, pp. 205–224, 1991.

[35] D. M. Titterington, "Common structure of smoothing techniques in statistics," *Internat. Statist. Rev.*, vol. 53, no. 2, pp. 141–170, 1985.

[36] J. H. Friedman, "Regularized discriminant analysis," *J. Amer. Statist. Assoc.*, vol. 84, no. 405, pp. 165–175, 1989.

[37] M. Skurichina, S. Raudys, and R. P. W. Duin, "K-nearest neighbors directed noise injection in multilayer perceptron training," *IEEE Trans. Neural Networks*, vol. 11, no. 2, pp. 504–511, 2000.

[38] J. Sietsma and R. J. F. Dow, "Neural net pruning—why and how," in *Proc. IEEE International Conference on Neural Networks I*, pp. 325–333, San Diego, Calif, USA, 1988.

[39] V. N. Vapnik and A. Ya. Chervonenkis, *Theory of Pattern Recognition. Statistical Problems of Learning*, Nauka, Moscow, Russia, 1974.

[40] V. N. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer-Verlag, New York, NY, USA, 1982.

[41] A. Kolmogorov, "Three approaches to the quantitative definition of information," *Problemy Peredachi Informatsii*, vol. 1, no. 1, pp. 3–11, 1965.
[42] J. Rissanen, "Stochastic complexity and modeling," *Ann. Statist.*, vol. 14, no. 3, pp. 1080–1100, 1986.
[43] I. Tabus, J. Rissanen, and J. Astola, "Classification and feature gene selection using the normalized maximum likelihood model for discrete regression," *Signal Process.*, vol. 83, no. 4, pp. 713–727, 2003, Special issue on Genomic Signal Processing.
[44] T. M. Cover and J. M. van Campenhout, "On the possible orderings in the measurement selection problem," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, no. 9, pp. 657–661, 1977.
[45] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Classification, Pattern Recognition and Reduction of Dimensionality*, P. R. Krishnaiah and L. N. Kanal, Eds., vol. 2 of *Handbook of Statistics*, pp. 835–855, North-Holland Publishing, Amsterdam, The Netherlands, 1982.
[46] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inform. Theory*, vol. 14, no. 1, pp. 55–63, 1968.
[47] P. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vol. 26, no. 9, pp. 917–922, 1977.
[48] Y. Hamamoto, S. Uchimura, Y. Matsuura, T. Kanaoka, and S. Tomita, "Evaluation of the branch and bound algorithm for feature selection," *Pattern Recognition Lett.*, vol. 11, no. 7, pp. 453–456, 1990.
[49] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.
[50] A. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 2, pp. 153–158, 1997.
[51] S. Ross, *A First Course in Probability*, Macmillan, New York, NY, USA, 4th edition, 1994.
[52] B. Efron, *The Jackknife, the Bootstrap and Other Resampling Plans*, vol. 38 of *CBMS-NSF Regional Conference Series in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 1982.
[53] B. Efron, "Estimating the error rate of a prediction rule: improvement on cross-validation," *J. Amer. Statist. Assoc.*, vol. 78, no. 382, pp. 316–331, 1983.
[54] M. R. Chernick, *Bootstrap Methods. A Practitioner's Guide*, John Wiley & Sons, New York, NY, USA, 1999.
[55] U. Braga-Neto and E. R. Dougherty, "Bolstered error estimation," *Pattern Recognition*, vol. 37, no. 6, pp. 1267–1281, 2004.
[56] C. Sima, U. Braga-Neto, and E. R. Dougherty, "Superior feature-set ranking for small samples using bolstered error estimation," to appear in Bioinformatics, 2005.

Ulisses Braga-Neto: Section of Clinical Cancer Genetics, The University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA; Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA

Email: ulisses_braga@yahoo.com

Edward R. Dougherty: Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA; Department of Pathology, The University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA

Email: edward@ee.tamu.edu

4 Clustering: revealing intrinsic dependencies in microarray data

Marcel Brun, Charles D. Johnson, and Kenneth S. Ramos

4.1. Introduction

Informal definitions for clustering can be found in the literature: the process of "unsupervised classification of patterns into groups" [1], the act of "partitioning of data into meaningful subgroups" [2], or the process of "organizing objects into groups whose members are similar in some way" [3]. In the context of pattern recognition theory, the objects are represented by vectors of *features* (the measurements that represent the data), called *patterns*. With these concepts in mind, clustering can be defined as the process of *partitioning the vectors into subgroups whose members are similar relative to some distance measure*. Therefore, two key questions that must be addressed prior to cluster implementation are about the distance to be used to measure the similarity of the objects and how to form the partitions that best group together these objects.

The answer to the first question depends on each particular problem where clustering is applied. The distance between patterns should reflect the relation that is considered significant for the analysis. The rationale for each distance measure will be addressed in this chapter. The second question relates to computational efficiency considerations and criteria to evaluate the quality of clustering. This too is dependent on the question being proposed.

The chapter is divided into four sections with several examples at the end. The section on Clustering Microarray Data introduces the application of clustering to microarray data, illustrating the practical aspects of these techniques. Measures of Similarity develops the topic of distance measures. The next section, Clustering Algorithms, presents the implementation of popular algorithms and their applicability to microarray data analysis. Lastly, the final section, Interpretation and Validation, discusses the available procedures to measure the validity of the resulting partitions, showing several examples of clustering applied to microarray data to solve specific biological questions.

4.2. Clustering microarray data

Data clustering has been used for decades in image processing and pattern recognition [4], and in the last several years it has become a popular data-analysis

technique in genomic studies using gene-expression microarrays [5, 6, 7, 8]. Each microarray provides expression measurements for thousands of genes, and clustering is a useful exploratory technique to analyze gene-expression data as it groups similar genes together and allows biologists to identify potentially meaningful relationships between them and to reduce the amount of information that must be analyzed. The function of genes could be inferred through "guilt by association" or appearance in the same cluster. Genes clustered together could have related functions or be coregulated (as demonstrated by other evidence such as common promoter regulatory sequences and experimental verification). Often, there is the additional goal of identifying a small subset of genes that are most diagnostic of sample differences. Time-series clustering groups together genes whose expression levels exhibit similar behavior through time, with similarity considered suggestive of possible coregulation. Another common use of cluster analysis is the grouping of samples (arrays) by relatedness in expression patterns. The expression pattern is effectively a complex phenotype and cluster analysis is used to identify samples with similar and different phenotypes. In medical research, this approach allows the discrimination between pathologies based on differential patterns of gene-expression, rather than relying on traditional histological methods. For instance, Eisen et al. [5] used cluster analysis to identify genes that show similar expression patterns over a wide range of experimental conditions in yeast, and Alizadeh et al. [9] used cluster analysis to identify subsets of genes that show different expression patterns between different types of cancers.

The main assumption underlying unsupervised cluster analysis for gene-expression data is that genes that belong to the same biological process, and genes in the same pathway, would have similar expression over a set of arrays (be it time series or condition dependent). A large number of papers have been published describing algorithms for microarray data clustering [5, 10, 11, 12], but a few analyze the relationship between the algorithms and the information that is supposed to be derived from the analysis [13]. To better understand this problem, we can separate the use of clustering algorithms in microarray data analysis into two areas: visualization and class discovery.

4.2.1. Notation

The microarray data for a set of m experiments $S_1, \ldots, S_m$ and n genes $g_1, \ldots, g_n$ is usually represented by a two-dimensional matrix M, where the value $M(i, j)$ represents the expression level of gene g_i for the sample S_j. Each gene g_i corresponds to a row on the matrix M, and for simplicity of notation it can be represented by a vector $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$, where each value x_{ij}, $j = 1, \ldots, m$, represents the expression of the gene g_i for the sample S_j. A sample S_j corresponds to a column within the matrix M, and may be represented by vectors $\mathbf{X}_j = (X_{1j}, \ldots, X_{nj})$, where each value X_{ij}, $i = 1, \ldots, n$, represents the expression of the gene g_i for the sample S_j.

The expression of the gene g_i in the sample S_j may be represented alternatively by $M(i, j)$, x_{ij}, or X_{ij}. When the context is clear, the notation can be loosened, and genes and samples can be represented by vectors $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ and $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$, respectively.

A sample may represent different subjects. In some cases, the values $\mathbf{X}_j$ associated to the sample S_j may represent the intensity of one of the two fluorophores in cDNA array, or the intensity of only the single channel for an Affymetrix-type array. It may also represent the ratio of the intensities for two fluorophores on one array, or the ratio of the intensities of two single channel arrays. In some cases, a logarithmic transformation is applied, and $\mathbf{X}_j$ represents the logarithm of the values.

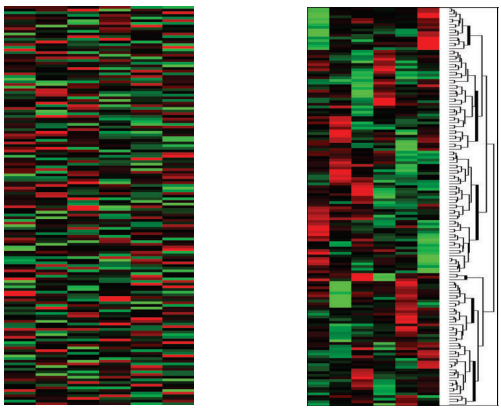
A *cluster* is sets of objects, which may be genes, $C = \{g_{i1}, \ldots, g_{ik}\}$, or samples, $C = \{S_{j1}, \ldots, S_{jk}\}$, depending on the type of analysis. A *clustering* of the data is a partition $\mathcal{C} = \{C_1, \ldots, C_K\}$ of the objects into K clusters. Each object belongs to one and only one set C_h with $h = 1, \ldots, K$. In an abuse of notation, sometimes a gene g_i will be denoted by its expression vector $\mathbf{x}_i$, and a sample S_j will be denoted by its expression vector $\mathbf{X}_j$. In that way, the expression "$\mathbf{x}_i \in C$" means really "$g_i \in C$." When the partition is of physical origin, thus representing the true clustering of the data, the sets are referred to as *classes* instead of clusters. For example, genes may be originally classified, thus partitioned, based on their role in the cell, as defined in the Gene Ontology Consortium [14].

4.2.2. Visualization

DNA microarrays are used to produce large sets of expression measurements and require efficient algorithms for visualization. Heat maps are used to represent thousands of values in a combined fashion. Most microarray experiments are used to study the relationship between biological samples, looking for genes that are differentially expressed. If X_i and Y_i represent the expression of the gene g_i in two different samples S_X and S_Y, then the ratio $T_i = X_i/Y_i$ between the two values gives the measure of difference between the two samples for gene g_i. For example, $T_i = 2$ indicates that gene g_i is expressed twice as high in sample S_X than in sample S_Y. A problem that arises with the use of ratios is that their interpretation is not symmetrical, that is, if gene g_i is twice as high in S_Y than in S_X, then $T_i = 0.5$. This limitation is commonly solved by applying a base 2 logarithm transformation to the ratio, so that equivalent fold changes in either direction have the same absolute value. In the previous example, $\log_2(2) = 1$ and $\log_2(0.5) = -1$, and the values are directly comparable. In addition, logarithmic transformation yields a near normal distribution of values which aids in subsequent statistical analyses [15].

The first efficient way to visualize microarray data [5] uses a 2-color representation, red for *up-regulated* genes, where the log of the ratio is positive, and green for *down-regulated* genes, where the log of the ratio is negative. The visualization consists of a matrix of colored cells, where each column represents a sample and each row represents a gene, and the brightness of the cell is proportional to the log of the ratio (Figure 4.1a). This visualization is usually accompanied by the name of the genes and the name of the samples, and it helps to visually identify different expression patterns.

A great improvement of the resulting image involves the ordering of the genes, so that ones with similar *profile* (i.e., the same expression pattern across the



(a) (b)

Figure 4.1. Visualization of gene-expression: (a) unordered, (b) ordering by hierarchical clustering.

samples) are placed together. This is clearly a task for clustering algorithms, and the hierarchical algorithm, to be defined in the next section, is usually selected because it creates a natural ordering of the genes based on a measure of similarity. Figure 4.1b shows how three common patterns are immediately revealed once the genes are sorted by proximity using hierarchical clustering based on Euclidean distance between the profiles. Clustering can also be applied to the samples to visualize common patterns of gene signatures (the expression pattern for a sample across the genes). Finally, the genes and the samples can be sorted in a combined fashion, by simultaneous clustering of both genes and conditions. This approach allows a clear visualization of similarity in gene-expression based on a sub-subset of attributes or samples. Another way to visually display the data based on clustering analysis was presented by Rougemont and Hingamp [16], where network analysis techniques are combined with correlation-based clustering to study DNA microarray data.

4.2.3. Class discovery

Gene-expression profiles refer to the expression values for a particular gene across various experimental conditions (or many genes under a single experimental condition). It is a key step in the analysis to reveal the responsiveness of genes (profiling), and discovering new classes of genes for classification (taxonomy).

These approaches are limited by the large number of variables (or genes), limited sample sizes, limited knowledge of the complete function of many genes, and the lack of knowledge of the underlying classes or subclasses.

In this process, clustering techniques may be used to identify unrecognized tumor subtypes, clustering algorithms are applied to the data to group the samples,

based on similarity of gene-expression. If an initial partition agrees with prior biological understanding, further refining (subpartitions) may reveal unknown subclasses, for example, in cancer [9, 17, 18, 19, 20]. The discovery of new classes can then be used as input for the supervised training of a classifier, after biological analysis of the validity of the new classes.

4.3. Clustering algorithms

4.3.1. Measures of similarity

A key concept in all clustering algorithms is the similarity between the objects to be grouped. Once the objects (genes or samples) are represented as vectors in a high-dimensional space (vectors $\mathbf{x}$ or $\mathbf{X}$, resp.), the concept of similarity is naturally based on distances between the vectors. If two objects are similar, it is expected that their vector representations are also similar, so the distance between them should be small. The distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is usually denoted by $d(\mathbf{x}, \mathbf{y})$. A brief list of distance measures include the Euclidean distance, defined by

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}, \tag{4.1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the vectors of length m, representing two objects. The Euclidean distance is a particular form of the more general distance measures and can be represented in the form

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{m} (x_i - y_i)^p \right)^{1/p}, \tag{4.2}$$

for $p = 2$. The case when $p = 1$ gives the city-block metric

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} |x_i - y_i|. \tag{4.3}$$

Distance measures can be replaced by similarity functions. Some of the usual functions are the *Pearson's correlation*

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{m} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m} (x_i - \overline{x})^2 \sum_{i=1}^{m} (y_i - \overline{y})^2}}, \tag{4.4}$$

where $\overline{x}$ and $\overline{y}$ are the average values of the vectors $\mathbf{x}$ and $\mathbf{y}$, respectively, and the noncentered Pearson's correlation

$$\rho'(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{m} (x_i)(y_i)}{\sqrt{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i^2}} \tag{4.5}$$

that represents the cosine of the angle between the two vectors [21]. Some algorithms can use both distance measures and similarity functions.

When the algorithm does not allows the use of similarity functions, they can be converted to dissimilarity functions (that may be nonmetric) via simple transformation. For example, the Pearson's correlation is constrained always in the range $[-1, 1]$, allowing it to be transformed in a dissimilarity function via two transforms $d_1(\mathbf{x}, \mathbf{y}) = 1 - \rho(\mathbf{x}, \mathbf{y})$ and $d_2(\mathbf{x}, \mathbf{y}) = 1 - |\rho(\mathbf{x}, \mathbf{y})|$. In the first case, the distances are in the range $[0, 2]$, and vectors whose values are directly or inversely related have a large distance. In the second case, the range is $[0, 1]$, and vectors whose values are inversely related have a small distance. Other measures of similarity can be used; depending on the underlying assumption of what the meaning of similarity is.

The selection of a measure of similarity should be based on two considerations: the clustering algorithm and the biological meaning of the word "close." As an example, correlation-based distance may bring together genes whose expression is different, but have a similar behavior, and which would be considered "different" by Euclidean distance. Some algorithms, like hierarchical, allow one to use any distance measure, but others are strongly related to a specific distance. The k-means algorithm, for example, minimizes the Euclidean distance between the vectors and the centroids of the clusters.

4.3.2. Preprocessing

Preprocessing of the data is an important step prior to clustering. The large number of genes present in a microarray experiment may be excessive for application of some algorithms with limited resources. As many of the algorithms are based on Euclidean distance between samples, the first step should consist of normalization to avoid samples with the larger dynamic range to take over the process. A good review of normalization of microarray data can be found in [22]. A second step is the removal of all genes that show low variation across samples, which may affect negatively the clustering process.

Researchers may be tempted to apply principal component analysis (PCA) to reduce the dimensionality of the data prior to clustering, but it is not proved to improve the results. It has been suggested that (a) the quality of the clusters is not necessarily higher with PCA than with the whole dataset and (b) in most cases the first principal components does not yield the best clusters [23].

4.3.3. Clustering

A way to classify the clustering algorithms is based on how the algorithm forms the groups: *hierarchical algorithms* work on successive splitting (*divisive clustering*) or merging (*agglomerative clustering*) of the groups, depending on a measure of distance or similarity between objects, to form a hierarchy of clusters, while *partitioning algorithms* search for a partition of the data that optimizes a global measure of quality for the groups, usually based on distance between objects. Hierarchical

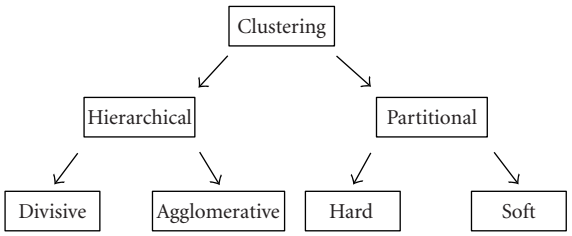


Figure 4.2. A basic taxonomy of clustering algorithms.

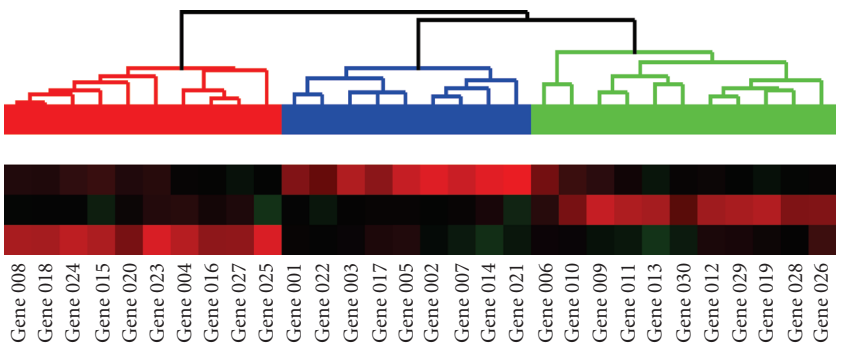


Figure 4.3. Hierarchical tree resulting from clustering.

algorithms are also classified by the way the distances or similarities are updated (*linkage*) after splitting or merging of the groups, which has a great influence on the resulting clustering. Hierarchical algorithms can generate partitions of the data as well, and are extensively used for this purpose, because each level of the hierarchy is a partition of the data.

Another way to classify the algorithms is based on their output: in *hard clustering* the output is a partition of the data, while in *soft* (i.e., fuzzy) *clustering* the output is a membership function, so each pattern can belong to more than one group, with some degree of membership. A fuzzy cluster defines naturally a partition of the data, defined by the maximum membership of each object. The quality of a clustering algorithm is often based on its ability to form meaningful partitions, Figure 4.2 shows a simple taxonomy of clustering algorithms.

The selection of a particular algorithm should be strongly related to the problem at hand. Each algorithm has its own strengths and weaknesses, and is better adapted to a particular task. For example, hierarchical clustering algorithms are extremely powerful for exploratory data analysis because it does not need prior specification of the number of clusters, and their outputs can be visualized as a tree structure, called a dendrogram (Figure 4.3). On the other hand, when using partitioning techniques, the groups are usually defined by a representative vector, simplifying the description of the resulting clusters (Figure 4.4).

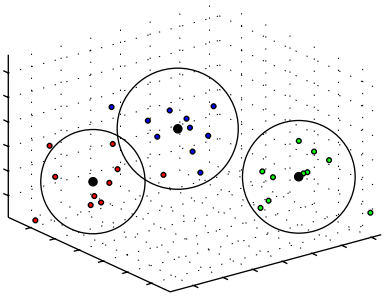


Figure 4.4. Centers of the clusters for a partitioning clustering (k-means).

Most of the partitioning algorithms are based on the minimization of an objective function that computes the quality of the clusters. The most common objective function is the squared error to the centers of the clusters. Let $\mathcal{C} = \{C_1, \ldots, C_K\}$ be a clustering of the data, and let μ_k be a vector representing the center of the cluster C_k, for $k = 1, \ldots, K$. The objective function $\mathcal{J}$ is defined by

$$\mathcal{J} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} ||\mathbf{x} - \mu_k||^2. \tag{4.6}$$

The objective function $\mathcal{J}$ is the average square distance between each point and the center of the cluster where the point belongs. It can be interpreted also as a measure of how good the centers μ_k are as representatives of the clusters. One limitation of this objective function is the need of a cluster center to represent the points. Usually for this objective function, the centers are defined as the average of all points in the group

$$\mu_k = \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}, \tag{4.7}$$

where n_k is the number of points in cluster C_k. The objective function can be simplified, and made explicitly dependent only on the points and not the centers, by

$$\mathcal{J} = \frac{1}{2n^2} \sum_{k=1}^{K} n_k \sum_{\mathbf{x}, \mathbf{x}' \in C_k} \|\mathbf{x} - \mathbf{x}'\|^2. \tag{4.8}$$

In the last equation, the objective function $\mathcal{J}$ depends only on the square distance between the points in the clusters, and there is no need to compute their centers. The second sum can be considered as a measure of tightness of the clusters, and can be replaced by measures better suited for a particular problem [21].

Partitioning algorithms, based on the minimization of an objective, suffer two major drawbacks. The first is that they work well with similar size compact clusters, but often fail when the shape of the cluster is more complex, or when there is a large difference in the number of points between clusters. The second drawback is that the objective function decreases as a function of the number of clusters in a nested sequence of partitions (a new partition is obtained by splitting in two one cluster from the previous partition). Given this property, the best partitioning of the data would be when $K = n$ clusters and each point is a cluster by itself. To address this problem, either the number of classes must be known beforehand, or some additional criteria must be used that penalizes partitions with large numbers of clusters.

The objective function is only a measure of the quality of a partition of the data. The naive way to find the best partition is to compute the objective function for all possible partitions, and select the one that has a minimum value of objective function. The number of possible partitions grows exponentially with n, the size of the set, so it is unfeasible except for very small problems. The most often used approach is iterative approximation, starting with an initial partition and modifying it iteratively while reducing the objective function.

4.3.4. k-means

One of the most common iterative algorithms is the k-mean algorithm, broadly used because of its simplicity of implementation, its convergence speed, and the usually good quality of the clusters (for a limited family of problems).

The algorithm is presented with a set of n vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and a number K of clusters, and computes the centroids $\mu_1, \mu_2, \mu_3, \ldots, \mu_k$, that minimizes the objective function $\mathcal{J}$. One of the most used implementations of the algorithm, called Forgy's cluster algorithm, starts with a random election for the centroids, and then repeatedly assigns each vector to the nearest centroid and updates the centroids positions, until convergence is reached, when the update process does not change the position of the centroids. The procedure can be done in batch or iteratively. In the first case all the vectors are assigned to a centroid before the update is completed. In the second part, the centroids are updated after each assignment is made. Some variations of the algorithms have been described by Duda et al. [21], Gose et al. [24], and Theodoridis and Koutroumbas [25].

k-means is one of the simplest algorithms known to perform well with many datasets, but its good performance is limited mainly to compact groups. When the points are drawn from a mixture of Gaussians, the k-means algorithm is indeed a gradient descent algorithm that minimizes the quantization error [26]. As with many gradient descent algorithms, one drawback of the k-means algorithm is that it can reach a local minimum of the objective function, instead of the desired global minimum, meaning that convergence is reached but the solution is not optimal. As an example, there may be cases where the clusters are compact and well separated, but the algorithm is not able to separate them, usually because the starting points were inconveniently placed. Figure 4.5 shows an example where

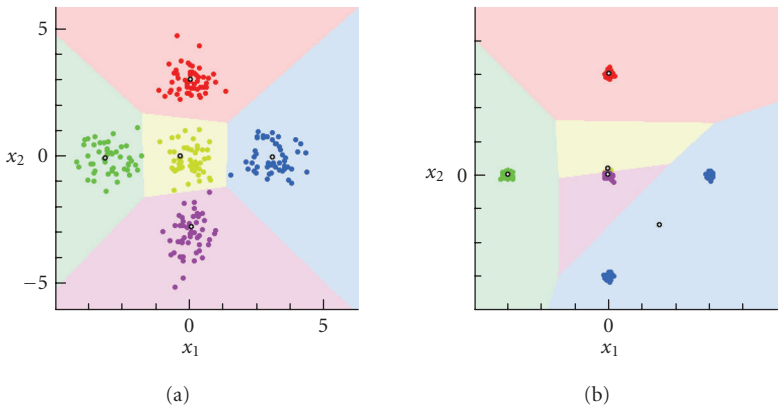


Figure 4.5. Example of k-means for two variables: x_1 and x_2. (a) When there is some dispersion in the groups; the centers of the cluster are able to skip to the right groups. (b) Tighter clusters impede the algorithm to work well, stopping at a local minimum of the objective function.

k-means performs better on disperse sets than in large ones. In Figure 4.5a the final position of the centroids is placed. An analysis of this problem is presented in [27]. One way to overcome this problem is by running the algorithm multiple times, with different random seeds, and then selecting the partition that appears with most frequency.

4.3.5. Fuzzy c-means

In the k-means algorithm, each vector is classified as belonging to a unique cluster (hard cluster), and the centroids are updated based on the classified samples. In a variation of this approach, known as *fuzzy c-means*, all vectors have a degree of membership (or a probability) of belonging to each cluster, and the respective centroids are calculated based on these probabilities. Let $P(\omega_k|\mathbf{x}_i)$ be the probability of the ith vector belonging to the kth cluster, it can be estimated from the data, based on the distances $d_{ik} = d(\mathbf{x}_i, \mu_k)$ and a constant parameter b that controls the "fuzziness" of the process (Duda et al. [21]):

$$P(\omega_k|\mathbf{x}_i) = \frac{(1/d_{ik})^{1/(b-1)}}{\sum_{r=1}^{K}(1/d_{ir})^{1/(b-1)}}. \tag{4.9}$$

Unlike the k-mean algorithm, where the center for each cluster is computed as the average of the vectors in that cluster, in fuzzy c-means the center μ_k of the kth cluster is calculated as a weighted average, using the probabilities as weights,

$$\mu_k = \frac{\sum_{i=1}^{n}\left[P(\omega_k|\mathbf{x}_i)\right]^b \mathbf{x}_i}{\sum_{i=1}^{n}\left[P(\omega_k|\mathbf{x}_i)\right]^b}. \tag{4.10}$$

As with k-means clustering, the process of assigning vectors to centroids and updating the centroids is repeated until convergence is reached.

4.3.6. Hierarchical clustering

Hierarchical clustering creates a hierarchical tree of similarities between the vectors, called a dendrogram. The most common implementation of this strategy is agglomerative hierarchical clustering, which starts with a family of clusters with one vector each, and merges the clusters iteratively based on some distance measure until there is only one cluster left, containing all the vectors. For a problem with n objects to be clustered, the algorithm starts with n clusters containing only one vector each, $C_i = \{\mathbf{x}_i\}$, $i = 1, 2, \ldots, n$. The initial distance between each pair of clusters is defined by the distance between their elements $d(C_i, C_j) = d(\mathbf{x}_i, \mathbf{x}_j)$. The algorithm repeatedly merges the two nearest clusters, and updates all the distances relative to the newly formed cluster, until there is only one cluster left, containing all the vectors.

Figure 4.6 shows an example of the complete process applied to 5 genes and 3 experiments (5 vectors of length 3). Initially there are 5 clusters, $C_1, \ldots, C_5$, each one containing one gene (Figure 4.6a). In the first step, as each cluster contains only one element, the distance between the clusters is defined as the Euclidean distances between the vectors that belong to them. The closest vectors are genes 1 and 4, so they are merged into a new cluster C_{14}, (Figure 4.6b). To continue the process, the distances between the unchanged clusters and the new cluster C_{14} are computed as function of their distance to C_1 and C_4. There is a need to compute the distances $d(C_2, C_{14})$, $d(C_3, C_{14})$, and $d(C_5, C_{14})$. Distances between nonchanging clusters (in this instance, genes 2, 3, and 5) do not need to be updated. Based on the new distances, a new pair of nearest clusters is selected. In this case clusters C_3 and C_5 are merged into a new cluster C_{35} (Figure 4.6c), and the new distances are computed for this new cluster relative to the unchanged clusters C_{14} and C_2. In the new set of distances, the nearest clusters are C_2 and C_{14}, so they are merged into a new cluster C_{124} (Figure 4.6d). Finally, the two remaining clusters, C_{35} and C_{124}, are merged into a final cluster, C_{12345}, that includes all five genes (Figure 4.6e). The dendrogram tree (Figure 4.6f) resumes the whole process. The length of the horizontal lines indicates the distance between the clusters.

The process does not define a partition of the system, but a sequence of nested partitions $\mathcal{C}_1 = \{C_1, C_2, C_3, C_4, C_5\}$, $\mathcal{C}_2 = \{C_{14}, C_2, C_3, C_5\}$, $\mathcal{C}_3 = \{C_{14}, C_2, C_{35}\}$, $\mathcal{C}_4 = \{C_{124}, C_{35}\}$, and $\mathcal{C}_5 = \{C_{12435}\}$, each partition containing one cluster less than the previous partition. To obtain a partition with K clusters, the process must be stopped K steps before the end. For example, stopping the process before the last merging ($K = 2$) will result in two clusters, C_{124} and C_{35} (Figure 4.6f). In the previous example, the distance between an existing cluster and a newly formed cluster was computed as the average distance between the vectors of both clusters. This is one way to update the distances, and a key point in the implementation because different updates lead to different results. Three common ways to update the distances are called *single*, *complete*, and *average* linkages.

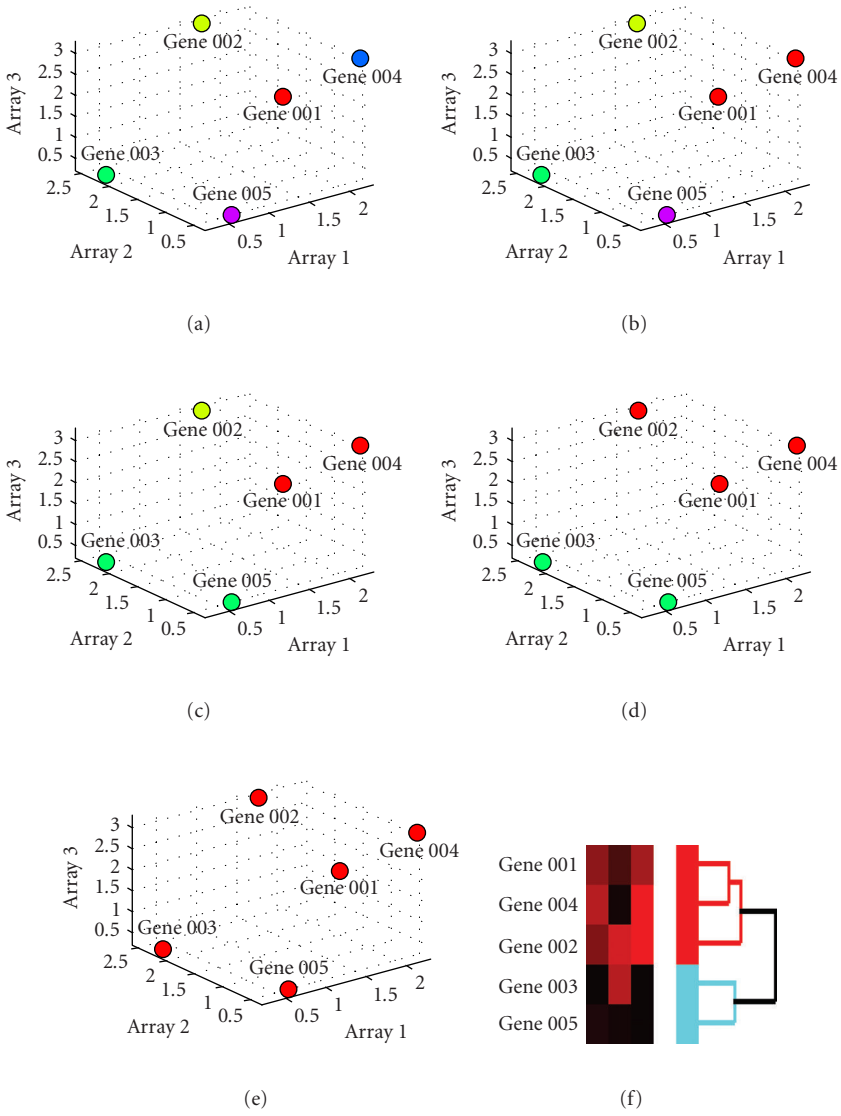


Figure 4.6. Example of agglomerative clustering using complete linkage.

(i) In single linkage, when two clusters are joined into a new cluster C_i, the distance between C_i and an existing cluster C_j is the minimum distance between the elements of C_i and C_j,

$$d(C_i, C_j) = \min_{\mathbf{x}\in C_i,\, \mathbf{y}\in C_j} \left(d(\mathbf{x}, \mathbf{y})\right). \tag{4.11}$$

(ii) In complete linkage, when two clusters are joined into a new cluster C_i, the distance between C_i and an existing cluster C_j is the maximum distance between the elements of C_i and C_j,

$$d(C_i, C_j) = \max_{\mathbf{x}\in C_i,\, \mathbf{y}\in C_j} \left(d(\mathbf{x}, \mathbf{y})\right). \tag{4.12}$$

(iii) In average linkage, when two clusters are joined into a new group C_i, the distance between C_i and an existing cluster C_j is the average distance between the elements of C_i and C_j,

$$d(C_i, C_j) = \frac{1}{n_i n_j} \sum_{\mathbf{x}\in C_i, \mathbf{y}\in C_j} d(\mathbf{x}, \mathbf{y}), \tag{4.13}$$

where n_i and n_j are the number of elements of clusters C_i and C_j, respectively.

Different linkages lead to different partitions, so that the selection of the linkage must be determined by the type of data to be clustered. For instance, complete and average linkages tend to build compact clusters, while single linkage is capable to build clusters with more complex shapes, but is more likely to be affected by spurious data.

Figure 4.7 shows an example of the differences obtained with different linkages. Figures 4.7a and 4.7b show the result of partitioning a set of points set in three subsets using Euclidean-distance-based hierarchical clustering for complete and single linkages, respectively. Clearly, the single linkage was not able to properly separate the groups, leaving one cluster with only one point (upper left corner). On the other hand, Figures 4.7c and 4.7d show the results of the same two algorithms over a different dataset with two visible classes. In this case, given the nonround shape of one of the groups, complete linkage fails to properly identify the classes (Figure 4.7c), while single linkage performs the task properly (Figure 4.7d).

4.3.7. MCLUST

MCLUST [2, 11, 28] is a clustering algorithm that assumes that the data is a mixture of multivariate normal distributions, one for each cluster $k = 1, \ldots, K$, with mean values μ_k and covariance matrices Σ_k. The covariance matrices can be decomposed in terms of their eigenvalue decomposition

$$\Sigma_k = \lambda_k D_k A_k D_k^t. \tag{4.14}$$

The CLUST algorithm estimates the parameters using the expectation maximization (EM) algorithm. The estimation is done in a two-step process similar to k-means clustering. In the first step the probabilities are estimated conditioned to the actual parameters, assigning each vector to one cluster (model), while in the second step the parameters of the models are estimated within the new clusters. The process is iterated until there is no more significant change in the parameters.

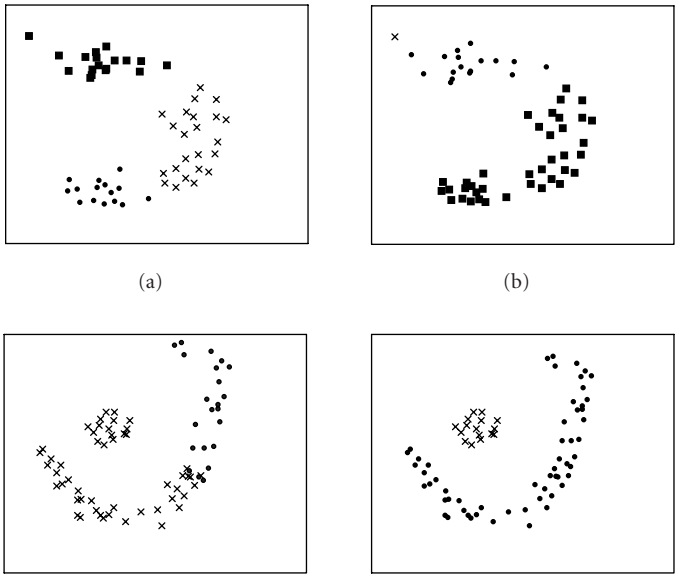


Figure 4.7. (a) and (c) Complete linkage. (b) and (d) Single linkage.

The result is an estimated set of K multivariate distributions, each one defining a cluster, and each vector assigned to the cluster with maximum conditional probability. Different assumptions on the model result in different constraints on the covariance matrices.

(i) Equal volume spherical: $\Sigma_k = \lambda I_d$ (where I_d is the identity matrix). The covariance matrices are all identical, diagonal, with the same value in the diagonal. The Gaussians are spherical.

(ii) Unequal volume spherical: $\Sigma_k = \lambda_k I_d$. The covariance matrices are all diagonal with the same value in the diagonal, but they can be different. The Gaussians are spherical, but they may have different volumes.

(iii) Elliptical: $\Sigma_k = \lambda DAD^t$. Each Gaussian is elliptical, but all have the same volume, shape, and orientation.

(iv) Unconstrained: there is no constraint for Σ_k. The Gaussians may have elliptical shape and different volumes.

Less constraints in the covariance matrices allow more flexibility to the model, but at the cost of more parameters to be estimated, which can increase the number of samples needed for a good estimation.

4.3.8. Graph-based algorithms

Another approach to clustering is based on graph theory. In this context, a clustering $\mathcal{C} = \{C_1, \ldots, C_K\}$ is represented by a weighted graph (V, ω), where the vertices

in V are the elements to be clustered and the weights $\omega(\mathbf{x}, \mathbf{y})$ indicate the similarity between the elements $\mathbf{x}$ and $\mathbf{y}$. The clustering problem consists in the specification of a graph (V, E), where the edges are based on ω, and such that highly similar elements belong to the same subgraph and dissimilar elements belong to different subgraphs [29].

Hartuv and Shamir [30] present a graph algorithm called highly connected subgraphs (HCS). The algorithm starts with the full graph $G = (V, E)$, where the edges are defined by the pairs $(\mathbf{x}, \mathbf{y})$ such that $\omega(\mathbf{x}, \mathbf{y})$ is greater than some threshold t, and identify HCS using minimum cuts recursively. The process can be resumed in the following steps:

(i) if G is highly connected,
  (a) return G as a cluster
(ii) if G is not highly connected,
  (a) find minimum cut for G and split it in subgraphs H1, H2,
  (b) process H1,
  (c) process H2,

where a graph is considered highly connected if the edge connectivity (minimum number of edges that need to be removed to disconnect the graph) is greater than half the number of vertices of the graph. A graph with only one element is considered highly connected. This algorithm returns clusters by recursively partitioning the graph, until highly connected graphs are reached. Because one-element graphs are considered highly connected, the algorithm will always stop.

In the extreme case, all the cluster will consist of a single element (if the algorithm cannot find highly connected graphs of size bigger than one), or all the set of vertices V will belong to the same cluster (if the original graph is highly connected). These extreme cases depend strongly on the election of the parameter t: a low value of t will generate a high amount of edges, so it is highly probable that the initial graph is highly connected, and a high value of t will generate a low amount of edges, so it is difficult to find highly connected subgraphs.

This technique is related to hierarchical clustering in the fact that it is a partitioning algorithm, differing only on the linkage process: a minimum cut is used for splitting, not based on a similarity distance between the clusters, and the process is stopped when high connectivity is reached, so it may not always provide a complete hierarchical tree.

Other algorithm presented by Sharan and Shamir [31], called cluster identification via connectivity kernels (CLICK), is directly derived from the HCS algorithm.

4.3.9. CAST

Cluster Affinity Search Clustering (CAST) is another graph-theory-based algorithm [6], where a kernel is initially defined and the clusters are grown based on average similarity between unassigned nodes and the existing clusters.

The algorithm CAST builds a graph based on a similarity measure $S(\mathbf{x}, \mathbf{y})$ between the elements and a threshold t that determines which values of the similarity are significant. The average affinity of an element $\mathbf{x}$ relative to a cluster C is given by

$$a(\mathbf{x}) = \frac{1}{|C|} \sum_{\mathbf{y} \in C} S(\mathbf{x}, \mathbf{y}). \tag{4.15}$$

The process can be resumed in the following steps.

(i) Start with an empty collection of clusters $C = \emptyset$, and a list $U = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of unused elements.
  (a) Initiate a cluster C_0 with one of the unused elements in U.
    (1) Add to C_0 the element of U with highest average affinity to C_0, if its affinity is greater than t. Upgrade the affinity values.
    (2) Remove from C_0 the elements with lowest average affinity to C_0, if it is smaller than t. Upgrade the affinity values.
    (3) Repeat until not changes are made.
  (b) Add the cluster C_0 to the clustering C.
  (c) Repeat until there are not any unused elements.

The CAST algorithm returns a clustering C where the clusters contain elements with high affinity, and it is inspired on a probabilistic model on graphs. The results are strongly dependent on the parameter t: CAST returns clusters where the affinity between any elements and its cluster is not smaller than t, but does not guarantee that each element is assigned to the cluster to which it has the largest affinity. Finally, unlike the other previously defined algorithms, the number of clusters cannot be defined beforehand.

4.3.10. Other algorithms

The previous reference list is not exhaustive. Other algorithms used for clustering of gene-expression data include the following.

(i) *SOM*, a clustering technique based on self organizing maps [10, 32], where the clusters are defined by the points of a grid that is adjusted to the data. Usually, the algorithm uses a two-dimensional grid in the higher-dimensional space.

(ii) *Simulated annealing* iteratively minimizes the Euclidean distance between elements in the same cluster, using a simulated annealing algorithm, and guarantees to eventually find the local minimum of the objective function [33].

Many good references can be found regarding clustering algorithms, involving mathematical and algorithmical aspects of the problem [4, 21, 34].

Each algorithm is proposed as a solution to some limitations of other algorithms, but except for the mixture models, which have a strong mathematical foundation including the estimation of the optimal number of clusters, they are usually heuristic techniques, and the results may be interesting and consistent. In the discussion between hierarchical clustering and partitioning algorithms, the former are well suited for analyses of hierarchies in the data, but are usually less

robust than partitioning algorithms. The MCLUST algorithm has the nice property of being based on statistical parameter estimation, but the quality of the results depends on the validity of the Gaussian assumptions. As with many model-based approaches, the result may only be as good as the model is for the data. If the data does not satisfy a Gaussian mixture model, some transformations may be applied to standardize it [11].

4.4. Interpretation and validation

Bring out the biologist! Once a clustering algorithm has grouped similar objects (genes and samples) together, the biologist is then faced with the task of interpreting these groupings (or clusters). For example, if a gene of unknown function is clustered together with many genes of similar, known function, one might hypothesize that the unknown gene has a related function. Alternatively, if biological sample "x" is grouped with other samples that have similar states or diagnoses, one might infer the state or diagnosis of sample "x." However, before subsequent work is completed to confirm or reject a hypothesis or, more importantly, to make a diagnosis based on the results of cluster analysis, several critical questions need to be asked. The first is how reproducible are the clustering results with respect to remeasurement of the data. Also, what is the likelihood that the grouping of the unknown samples or genes of interest with other known samples or genes is false (due to noise in the data, inherent limitations of the data, or limitations in the algorithm)? From a biological standpoint, clustering can work well when there is already a wealth of knowledge about the pathway in question, but it works less efficiently when this knowledge is sparse. The real question is whether the clusters make any biological sense. Only the biologist can answer this question.

4.4.1. Types of validation

Clustering is usually defined as a process that *aims to group similar objects* or as *unsupervised learning*. An open problem with clustering algorithms is the validation of results. As a data mining tool, a clustering algorithm is good if it generates new testable hypotheses, but as an analysis tool, it should be able to generate meaningful results that can be related to properties of the objects under study. There are two basic ways to compute the quality of a clustering algorithm. The first one is based on calculating properties of the resulting clusters, such as compactness, separation, and roundness. This is described as *internal validation* because it does not require additional information about the data. The second is based on comparisons of the partitions, and can be applied to two different situations. When the partitions are generated by the same algorithm with different parameters, it is called *relative validation*, and it does not include additional information. If the partitions to be compared are the ones generated by the clustering algorithm and the true partition of the data (or a subset of the data), then it is called *external validation*. External and relative validations are mainly based on comparison between

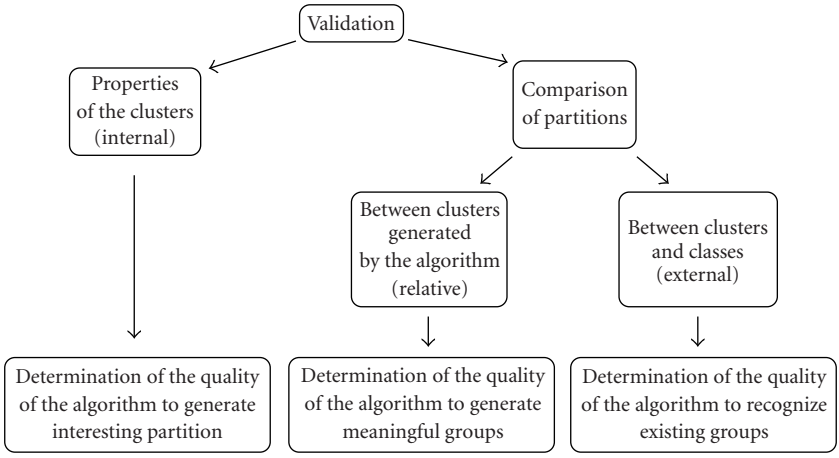


Figure 4.8. A simplified classification of validation techniques.

different partitions of the data. Figure 4.8 shows a hierarchy of validation techniques.

4.4.2. Internal validation

For internal validation, the evaluation of the resulting clusters is based on the clusters themselves, without additional information or repeats of the clustering process. This family of techniques is based on the assumption that the algorithms should search for clusters whose members are close to each other and far from members of other clusters.

The *Dunn's validation index* [35, 36] is defined as the ratio between the minimum distance between two clusters and the size of the larger cluster. Let $\mathcal{C} = \{C_1, \ldots, C_K\}$ a partition of the samples into K clusters:

$$V(\mathcal{C}) = \frac{\min_{h,k=1,\ldots,K,\, h\neq k} d(C_k, C_h)}{\max_{k=1,\ldots,K} \Delta(C_k)}, \tag{4.16}$$

where $d(C_k, C_h)$ is the distance between the two clusters and $\Delta(C_k)$ is the size of the cluster C_k. The value of $V(\mathcal{C})$ depends on the selection of the distance measure between clusters and the measure used for cluster size. Some basic examples of distance measures are the minimum, maximum, and average distances, as defined by equations (4.11), (4.12), and (4.13), respectively. There are many selections for the measure of the cluster size, some of them are as follows:

(i) maximum distance between two points:

$$\Delta(C) = \max_{\mathbf{x},\mathbf{y}\in C} d(\mathbf{x}, \mathbf{y}), \tag{4.17}$$

(ii) twice average distance to the centroid:

$$\Delta(C) = \frac{2}{|C|} \sum_{\mathbf{x} \in C} d\left(\mathbf{x}, \frac{1}{|C|} \sum_{\mathbf{x} \in C} \mathbf{x}\right). \tag{4.18}$$

More options for these measures can be found in [35, 36]. Each combination of distance measure and cluster size measure defines a different Dunn's index. In order to obtain meaningful results, it is important to select the measures that are more closely related to the problem. For example, twice average distance to the centroid penalizes clusters that are not round. This measure may assign lower scores to clusters obtained by single linkage hierarchical clustering than the ones obtained by average linkage hierarchical clustering, regardless of the ability of the clusters to explain the underlying biological process.

Other indices that can be computed are the *Davies-Bouldin* and *Silhouette* [35, 36, 37], *root mean square standard deviation*, and *R-squared* [38]. In [39], the dendrogram information is used to compute the *cophenetic correlation coefficient* (CPCC), which measures the proximity level for pairs of points based on the hierarchy tree. This index is applicable when using hierarchical clustering.

Another approach to internal validation is comparing the clusters with the distance between the points. This comparison is similar to the use of *Hubert's statistics* in external validation (below), but replacing the true partition matrix by the proximity matrix $d(i, j) = d(\mathbf{x}_i, \mathbf{x}_j)$ between points [39]. Finally, other indices are *compactness* and *isolation* of the clusters [40], and separation and homogeneity [31].

4.4.3. Measures of similarity between partitions

Assume that there exists two partitions of the same set of n objects into K groups: $\mathcal{C}^A = \{C_1^A, \ldots, C_K^A\}$ and $\mathcal{C}^B = \{C_1^B, \ldots, C_K^B\}$. Each element C_k^A and C_k^B of $\mathcal{C}^A$ and $\mathcal{C}^B$ is called a *cluster* and is identified by its index k. Let $k = I_A(\mathbf{x})$ be the index of the cluster to which a vector $\mathbf{x}$ belongs for the partition $\mathcal{C}^A$ (e.g., if $I_A(\mathbf{x}) = 3$, then the vector $\mathbf{x}$ belongs to the cluster C_3^A). The natural measure of disagreement (or error) between the two partitions is the error measure $\varepsilon(\mathcal{C}^A, \mathcal{C}^B)$ defined as the proportion of objects that belongs to different clusters,

$$\varepsilon(\mathcal{C}^A, \mathcal{C}^B) = \frac{\left|\{\mathbf{x} : I_A(\mathbf{x}) \neq I_B(\mathbf{x})\}\right|}{n}, \tag{4.19}$$

where $|S|$ indicates the number of elements of the set S.

The first observation that can be derived is that if the partitions are the same, but the order of the indices is changed in one of them (e.g., if $C_1^B = C_2^A$, $C_2^B = C_1^A$, $C_3^B = C_3^A, \ldots, C_K^B = C_K^A$), then, for any vector $\mathbf{x}$ in C_1^A, $I_A(\mathbf{x}) = 1 \neq 2 = I_B(\mathbf{x})$, and the error $\varepsilon(\mathcal{C}^A, \mathcal{C}^B)$ is greater than zero. It is clear that the disagreement between two partitions should not depend on the indices used to label their clusters.

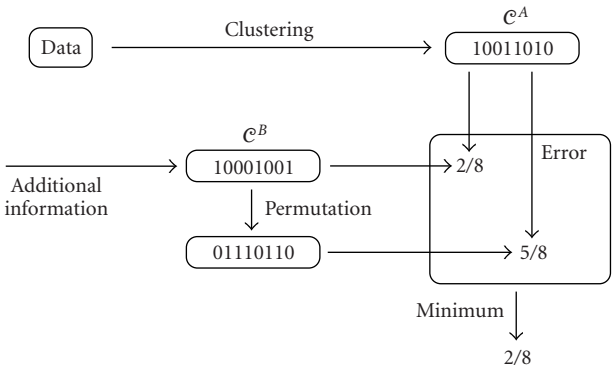


Figure 4.9. Schematic computation of misclassification for two-sets partitions.

A corrected measure, called misclassification rate, is defined by

$$\varepsilon^*(\mathcal{C}^A, \mathcal{C}^B) = \min_{\pi} \varepsilon(\mathcal{C}^A, \pi(\mathcal{C}^B)) \tag{4.20}$$

over all of the possible permutations π of the K sets in $\mathcal{C}^B$. A nice property of this measure is that if the data is modeled as a random labeled point process, $\mathcal{C}^B$ is the true partition of a set generated by the process and $\mathcal{C}^B$ is the result of a clustering algorithm, then this error measure is an estimator of the true error of the algorithm [41]. The application of this measure is limited by the fact that the number of permutations π for K clusters is equal to the factorial of K. For example, for 10 clusters there are 3628800 permutations to be analyzed. Therefore, except for a small number of clusters, direct computation of this measure using (4.20) is usually impractical. In such cases, suboptimal techniques may be used to find a-near-to-optimal permutation [27]. Figure 4.9 shows a schematic example on how to compute the misclassification rate, when there are 8 objects and two clusters, and the partition is represented by a vector of size 8, indicating the cluster label $I_A(\mathbf{x})$ for each object. In this example, $K = 2$ and there are only two permutations of the partition.

Another way to compare two partitions $\mathcal{C}^A$ and $\mathcal{C}^B$, without labeling the clusters (and therefore avoiding the permutation problem) is based on a pairwise comparison between the vectors. For each pair of vectors $\mathbf{x}$, $\mathbf{y}$ ($\mathbf{x} \neq \mathbf{y}$), there are four possible situations:

(a) $\mathbf{x}$ and $\mathbf{y}$ fall in the same cluster in both $\mathcal{C}^A$ and $\mathcal{C}^B$,
(b) $\mathbf{x}$ and $\mathbf{y}$ fall in the same cluster in $\mathcal{C}^A$ but in different clusters in $\mathcal{C}^B$,
(c) $\mathbf{x}$ and $\mathbf{y}$ fall in different clusters in $\mathcal{C}^A$ but in the same cluster in $\mathcal{C}^B$,
(d) $\mathbf{x}$ and $\mathbf{y}$ fall in different clusters in both $\mathcal{C}^A$ and $\mathcal{C}^B$.

The measure of disagreement between $\mathcal{C}^A$ and $\mathcal{C}^B$ is quantified by the number of pairs of vectors that fall in situations (b) and (c). Let a, b, c and d be the number of pair of different vectors that belong to situation (a), (b), (c), and (d), respectively,

and let $M = n(n-1)/2$ be the number of pair of different vectors. The following indices measure the agreement between the two partitions [39]:

(i) Rand statistic,

$$R = \frac{a+d}{M}, \tag{4.21}$$

(ii) Jaccard coefficient,

$$J = \frac{a}{a+b+c}, \tag{4.22}$$

(iii) Folkes and Mallow (FM) index,

$$\text{FM} = \sqrt{\frac{a}{a+b}\frac{a}{a+c}}. \tag{4.23}$$

The differences between the indices are subtle. The Rand statistic measures the proportion of pairs of vectors that agree by belonging either to the same cluster (a) or to different clusters (d) in both partitions. The Jaccard coefficient measures the proportion of pairs that belong to the same cluster (a) in both partitions, relative to all pairs that belong to the same cluster in at least one of the two partitions ($a+b+c$). In both cases, the measure is a proportion of agreement between the partitions, but in contrast with the Rand statistic, the Jaccard coefficient does not consider the pairs that are separated (belong to different clusters) in both partitions (d). The FM index measures the geometric mean of the proportion of pairs that belong to the same cluster in both partitions (a), relative to the pairs that belong to the same cluster for each partition ($a+b$ for $\mathcal{C}^A$ and $a+c$ for $\mathcal{C}^B$). As with the Jaccard coefficient, the FM index does not consider the pairs that are assigned to different clusters by the two partitions. The three measures share the property that they are zero if there is no agreement between the partitions ($a = d = 0$), and they are one if the agreement is complete ($b = c = 0$), and intermediate values are used as a quantitative measure of agreement between partitions.

The previous indices are based on the counting of the number of pairs of vectors, that are placed on the same or different clusters, for each partition. For each partition $\mathcal{C}$ the relationship between two vectors, whether they belong to the same cluster or not, can be represented by a similarity matrix $d(i,j)$ defined by $d(i,j) = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster, and $d(i,j) = 0$ if they belong to different clusters. The advantage of using this matrix instead of the four numbers a, b, c, and d is that it allows additional comparisons: let $\mathbf{d}^A$ and $\mathbf{d}^B$ be the similarity matrices induced by two partitions $\mathcal{C}^A$ and $\mathcal{C}^B$, two similarity indices are computed as function of the correlation and the covariance of these matrices,

(i) Hubert Γ statistic:

$$\Gamma = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\mathbf{d}^A(i,j)\mathbf{d}^B(i,j), \tag{4.24}$$

(ii) normalized Γ statistic:

$$\Gamma = \frac{1}{M\sigma^A\sigma^B}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\mathbf{d}^A(i,j)-\mu_A\right)\left(\mathbf{d}^B(i,j)-\mu_B\right), \tag{4.25}$$

where μ^A, μ^B, σ^A, and σ^B are the respective sample mean and standard deviation of the values in the matrices $\mathbf{d}^A$ and $\mathbf{d}^B$. The Hubert statistic is based on the fact that the more similar the partitions, the more similar the matrices would be, and this similarity can be measured by their correlation.

In this context, with the similarity matrices $\mathbf{d}^A$ and $\mathbf{d}^B$ representing the partitions $\mathcal{C}^A$ and $\mathcal{C}^B$, with their values being only 1 and 0, and by the fact that the matrices are symmetric, the previously defined Rand statistic can be rewritten as

$$\begin{aligned}
R &= \frac{a+d}{M} = \frac{M-b-c}{M} = 1-\frac{b+c}{M} \\
&= 1-\frac{1}{M}\sum_{i=1}^{n} n-1\sum_{j=i+1} nI_{[\mathbf{d}^A(i,j)\neq\mathbf{d}^B(i,j)]} \\
&= 1-\frac{1}{M}\sum_{i=1}^{n} n-1\sum_{j=i+1} n\left|\mathbf{d}^A(i,j)\neq\mathbf{d}^B(i,j)\right| \\
&= 1-\frac{1}{M}\sum_{i=1}^{n} n-1\sum_{j=i+1} n\left(\mathbf{d}^A(i,j)\neq\mathbf{d}^B(i,j)\right)^2 \\
&= 1-\frac{1}{2M}\sum_{i=1}^{n} n\sum_{j=1} n\left(\mathbf{d}^A(i,j)\neq\mathbf{d}^B(i,j)\right)^2,
\end{aligned} \tag{4.26}$$

where $|\bullet|$ represents absolute value. Equation (4.26) shows that the Rand index is inversely proportional to the square of the Euclidean distance between the matrices $\mathbf{d}^A$ and $\mathbf{d}^B$. In a similar way, the Jaccard coefficient and the two ratios in the FM index are all proportional to the Euclidean distance restricted to specific sections of the matrices (resp., where both matrices are different from zero, where the first matrix is one, and where the second matrix is one).

As an example, assume that there are $n = 5$ points, labeled $x_1,\ldots,x_5$, and the partitions are defined by $\mathcal{C}^A = \{\{1,2,3\},\{4,5\}\}$ and $\mathcal{C}^B = \{\{1,3\},\{2,4,5\}\}$. Figures 4.10a and 4.10b show the similarity matrices $\mathbf{d}^A$ and $\mathbf{d}^B$, with the shaded region representing the region of interest (the matrix is symmetrical and the diagonal is always 1).

In the example, the numbers in bold indicate that there are differences between the two matrices. There are $M = 10$ pairs of different points (shaded region) and the values for a, b, c, and d are 2, 2, 2, and 4, respectively. The indices computed over this example are $R = 0.6$, $J = 0.33$, FM $= 0.5$, $\Gamma = 0.2$, and $\Gamma^* = 0.15$.

Additional measures of similarity between partitions are presented in [39], including *Davies-Bouldin index*, *root mean square standard deviation* (RMSSTD), *R-squared*, and *distance between two clusters*.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$
$\mathbf{x}_1$	1	**1**	1	0	0
$\mathbf{x}_2$	1	1	**1**	**0**	**0**
$\mathbf{x}_3$	1	1	1	0	0
$\mathbf{x}_4$	0	0	0	1	1
$\mathbf{x}_5$	0	0	0	1	1

(a)

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$
$\mathbf{x}_1$	1	**0**	1	0	0
$\mathbf{x}_2$	0	1	**0**	**1**	**1**
$\mathbf{x}_3$	1	0	1	0	0
$\mathbf{x}_4$	0	1	0	1	1
$\mathbf{x}_5$	0	1	0	1	1

(b)

Figure 4.10. (a) Similarity matrix for $\mathbf{d}^A$. (b) Similarity matrix for $\mathbf{d}^B$.

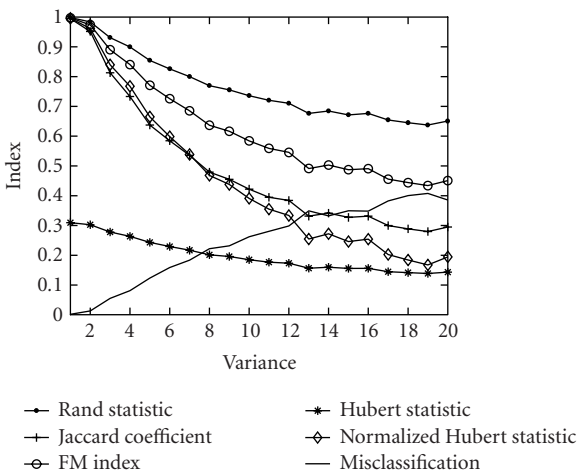


Figure 4.11. Indices and misclassification as function of the variance.

Once a similarity measure between partitions is defined, they can be used to measure the ability of a clustering algorithm to group the data in two ways: using the knowledge of the true partition (external validation), or comparing several results of the same algorithm (relative validation).

4.4.4. External validation

In *external validation*, the quality of the algorithm is evaluated by comparing the resulting clusters with prespecified information. In this case, $\mathcal{C}^A$ may be the result of the clustering algorithm and $\mathcal{C}^B$ may be the true partition of the data, if known, and the similarity between the partitions may be measured by the indices described previously [6, 27, 31, 38, 39, 42, 43].

Figure 4.11 shows an example of the indices as function of the variance in simulated data. The data was generated randomly using a mixture of three Gaussians

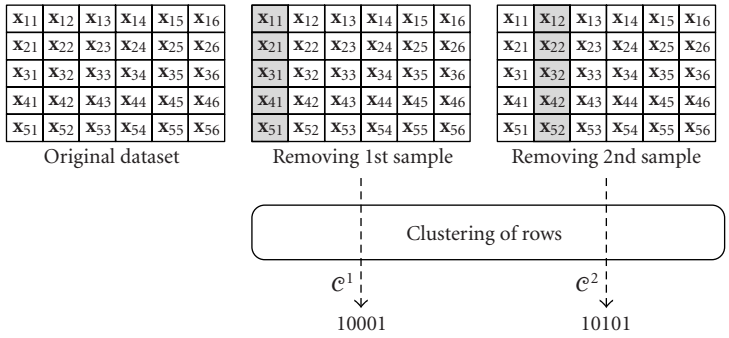


Figure 4.12. Clustering applied to a subset of the data.

with 10 points for each Gaussian, and variance varying from 1 to 20. The algorithm used is fuzzy c-means with $K = 3$, the true number of clusters. The indices were computed over the partition generated by the algorithm and the true partition of the data. The larger the variance, the greater the overlap between classes, and the more difficult it is for the algorithm to recognize them. In all cases, the indices decrease as the overlap of the clusters increases. Figure 4.11 also shows the misclassification rate for comparison with the indices. For very small variance, all the results of the clustering algorithm are perfect, the misclassification is zero, and all the indices are one, except Hubert's statistic.

The main drawback of this validation approach is that the classes must be known beforehand, so it seems counterintuitive to apply clustering when the result is already known. This limitation can be overcome in certain cases. In [40], the method is used with incomplete data, computing Hubert's statistics over the adjacency matrices to compare the true cluster (from annotation of functionality) with the results obtained from the algorithm. In [27], the quality of the algorithm was quantified using external validation on synthetic data generated from the clusters, based on the premise that the algorithm with better performance on the simulated data could be the best to be applied to the original data.

4.4.5. Relative validation

Usually, clustering algorithms are applied in cases where there is limited knowledge of the true classes in the data, and external validation is no longer a valid option. One way to overcome this problem is by measuring the consistency of the algorithms instead of the agreement with the true partitions. *Relative validation* consists of comparisons of the clusters obtained by the same algorithm, but under different conditions. There are mainly two ways to generate different conditions: using subsets of the data [43, 44] or changing the parameters of the algorithm [39].

In the first case, if there are m samples, the algorithm is run m times on the subsets obtained by removing one sample at a time. This approach is referred to as *leave-one-out validation*. Figure 4.12 shows an example with 5 objects ($n = 5$)

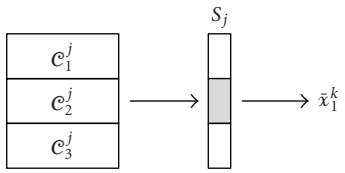


Figure 4.13. Example of computation of $\overline{\mathbf{x}}_j^k$.

and 6 samples ($m = 6$), where the clustering algorithm is applied to two subsets of the data, removing the first and second columns, obtaining two partitions $\mathcal{C}^1$ and $\mathcal{C}^2$, respectively. The process can be repeated for the six columns, obtaining six partitions of the data.

The result of this process is a family of partitions $\mathcal{C}^1, \ldots, \mathcal{C}^P$, each one computed over a slightly different dataset. The agreement between all these partitions gives a measure of the consistency of the algorithm and their predictive power (over the removed column) gives a measure of the ability of the algorithm to generate meaningful partitions.

Yeung's *figure of merit* (FOM) [43] is based on the assumption that the clusters represent different biological groups, and therefore, genes in the same cluster have similar expression profiles in additional samples. This assumption leads to a definition of the quality of a clustering algorithm as the spread of the expression values inside the clusters, measured on the sample that was not used for clustering. Let m be the number of samples, n the number of objects, and K the number of clusters. Let $\mathcal{C}^j = \{C_1^j, \ldots, C_K^j\}$ be the partition obtained by the algorithm when removing the sample S_j. The FOM for sample S_j is computed as

$$\mathrm{FOM}(K, j) = \sqrt{\frac{1}{n} \sum_{k=1}^{K} \sum_{i \in C_k^j} \left(\mathbf{x}_{ij} - \overline{\mathbf{x}}_j^k\right)^2}, \tag{4.27}$$

where $\overline{\mathbf{x}}_j^k$ is the jth element of the average of the vectors in C_k^j (see Figure 4.13).

The FOM (for the algorithm) is computed as the sum over the samples:

$$\mathrm{FOM}(K) = \sum_{j=1}^{m} \mathrm{FOM}(K, j). \tag{4.28}$$

If the clusters for a partition define compact sets of values in the removed sample, then their average distances to their centroids should be small. Yeung's FOM is the average measure of the compactness of these sets. The lower the FOM, the better the clusters are to predict the removed data and therefore, the more consistent the result of the clustering algorithm.

One of the drawbacks of this measure is that the decrease of the FOM as a function of the number of clusters may be mainly artificial, due to the fact that more clusters mean smaller average size for the clusters. In some situations,

the more clusters are present, the smaller the sets are. Sets with a smaller number of points are more likely to be compact. A solution to this problem is to adjust the values using a model-based correction factor $\sqrt{(n-K)/n}$. The result is called *adjusted FOM*. In practice, when n is large and K is small, as in clustering of microarray data, the correction factor is close to one and does not greatly affect the results.

Other ways to compare the resulting partitions have been described by Datta and Datta [44]. The approaches are based on the idea that the algorithms should be rewarded for consistency. As with the FOM measure, the algorithm is repeatedly applied to subsets of the data, where a different sample S_j is deleted for each subset, forming partitions $\mathcal{C}^j = \{C_1^j, \ldots, C_K^j\}$. Additionally, the algorithm is also applied to the whole set of samples, forming the partition $\mathcal{C}^0 = \{C_1^0, \ldots, C_K^0\}$. Let $C^j(i)$ be the cluster containing the vector $\mathbf{x}_i$ for the partition $\mathcal{C}^j$. Let $\overline{\mathbf{x}}_{C_i^j}$ be the average of the vectors in the cluster containing the vector $\mathbf{x}_i$ for the partition $\mathcal{C}^j$. The three measures of quality are defined as follows:

(i) *average proportion of non-overlap:*

$$V_1(K) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(1 - \frac{\left| C^j(i) \cap C^0(i) \right|}{\left| C^0(i) \right|}\right), \tag{4.29}$$

(ii) *average distance between means:*

$$V_2(K) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} d\left(\overline{\mathbf{x}}_{C_i^j}, \overline{\mathbf{x}}_{C_i^0}\right), \tag{4.30}$$

(iii) *average distance:*

$$V_3(K) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{\left| C^j(i) \right| \left| C^0(i) \right|} \sum_{\mathbf{x} \in C^j(i),\, \mathbf{y} \in C^0(i)} d(\mathbf{x}, \mathbf{y}). \tag{4.31}$$

Figure 4.14 shows a comparison of three clustering algorithms applied to a mixture of three Gaussians with 10 points for each Gaussian, all displaying indices and the true misclassification rate, averaged over 20 repetitions. The algorithms compared are hierarchical clustering (complete linkage, Euclidean distance), k-means, and fuzzy c-means, and the number of clusters varies from 2 to 7, and is shown in Figures 4.14a, 4.14b, and 4.14c, respectively. Almost all the indices show a change in slope (some show a local minimum) for $K = 3$, that is the correct number of classes in the data.

This is an example where without information about the real classes it is still possible to guess the correct number of clusters. This approach is not always valid, and depends strongly on the structure of the data and the existence, or not, of *true* classes within it. The graphs also show the relative agreement between misclassification rate (external measure) and the four indices, in particular for the average distance between means $V_2(K)$. The other indices do not increase for more than

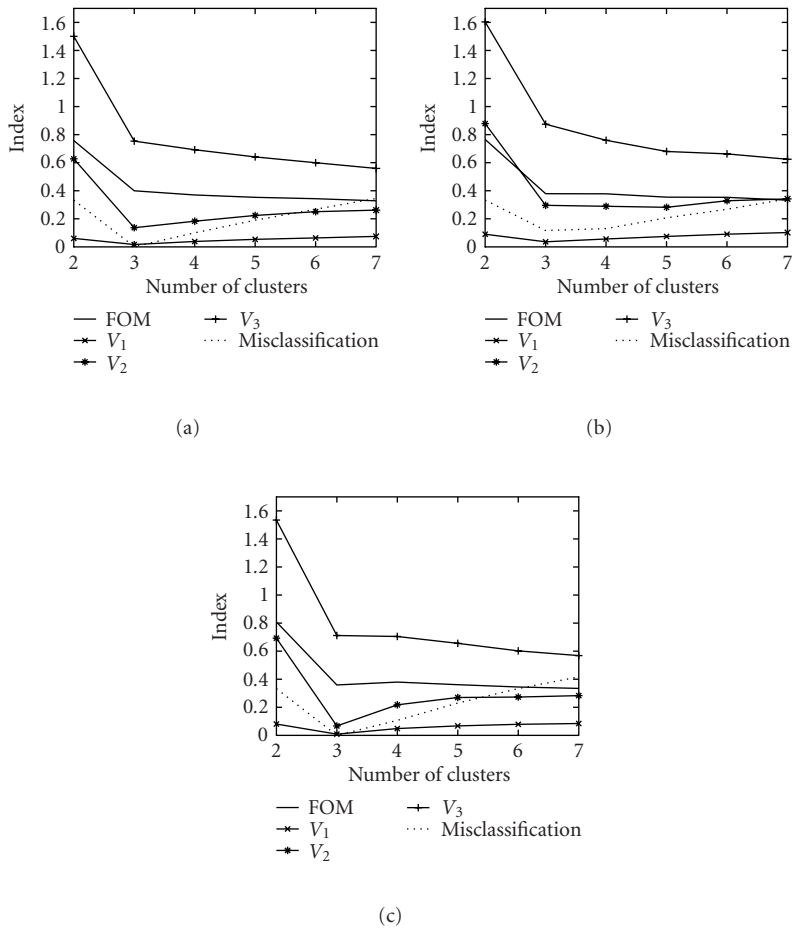


Figure 4.14. Indices for 3 different clustering algorithms: (a) hierarchical clustering, (b) k-means clustering, (c) fuzzy c-means clustering.

three clusters as they should. This may be explained by the fact that they are affected by the number of clusters independently of the real number of classes (in the extreme case with $K = n$ all indices are zero).

Relative validation techniques are a valid approach to compare clustering algorithms, guess the best number of clusters, and measure the quality of a single clustering algorithm. These measures do not indicate an agreement between the clusters and true classes in the data, but the ability of the algorithm to behave consistently over the data and their interpretation is related to the assumptions made in their definitions. For example, the FOM assumes a strong inter-dependence between samples, which is usually true in microarray data, but may be not true in processed data, like when PCA is used.

4.4.6. Statistical inference

Some studies analyze the validation of the clustering algorithms as a statistical inference process. Halkidi et al. [39] uses Monte Carlo techniques to test the null hypothesis of the random structure of the data. Given a set of points obtained from a mixture of known Gaussians, these investigators apply a clustering algorithm and the values of some external indices (like Hubert's and Rand). From the same model, using Monte Carlo techniques, the distribution of these indices under the null hypothesis is estimated and the null hypothesis is rejected (the clusters are better than random) if the index is above the 95% threshold. With this technique, the cluster structure obtained can be validated from the clustering algorithm. Also, Hubert's statistics can be used to compare the similarity matrix with a matrix containing cluster-based similarity between points, to validate hierarchical similarity-based algorithms. Dougherty et al. [27] present a model-based algorithm to find the best algorithm for a given problem. The method is based in the assumption that all genes with the same functionality have similar expression pattern, and this expression is modeled as a Gaussian distribution. Given the model, the algorithms can be compared based on their efficiency over simulation-based data using the model estimated from the original data. The algorithm with the lowest average error is assumed to be the best algorithm for the model. Different models may lead to different selection of the best algorithm. Kerr and Churchill [45] present a bootstrapping technique to validate clustering results. They create a number of simulated datasets based on the statistical model (estimated from the original data). "The match of a gene to a profile is declared 95% stable if it occurs in the analysis of the actual data and in at least 95% of the bootstrap clustering."

4.4.7. Other approaches

There are several techniques that cannot be classified in these three classes (external, internal, and relative). For example, the quality of the algorithms can be determined by their intrinsic properties as defined by criteria, such as *admission decision rules*, that are based on the shape and properties of the clusters they define [3, 46, 47]. Instead of looking at how well the algorithm performs on a particular dataset, this validation technique evaluates the algorithm in its capacity to solve a specific type of problems.

Attempts have been made to evaluate the validation measures as indicators of the quality of the clusters. Yeung et al. [43], for instance, used an external criterion to evaluate the validation measure by comparing the FOM (a relative criterion) to the adjusted rand index, computed between the clusters obtained by the algorithm and the real clusters.

A different perspective is used in the model-based algorithms, like in [43, 48, 49]. In these cases, there is an assumption of an underlying model, but this model is not used to find the best algorithm over a family of different techniques. Given the model, an algorithm is used to find its parameters: number, position, and shape of the clusters. Fraley and Raftery [28, 48] and Yeung et al. [43] use model-based

clustering to determine the structure of clustered data without using prior knowledge (except the fact that the model is a mixture of Gaussians), and present an alternative technique to compute the number of clusters. In this case, given a model of Gaussian mixtures for the data, explicit solution of the clustering problem can be obtained via EM algorithms. This is a clear example of how the best algorithm can be derived from a prior assumption in the model. The performance of the algorithms was also evaluated using synthetic datasets, showing that for model-based clustering the common agglomerative techniques have results similar to hierarchical techniques. As a drawback of the technique, the authors give examples where the assumption of Gaussian mixture models fails if the points are sampled from other distributions [49].

4.4.8. Discussion

There are excellent complete reviews of algorithms and validation measures [3, 39]. External criteria are useful if there exists previous knowledge about the structure of the data, as the real partitioning or an underlying distribution for some points. Internal and relative criteria are based on assumptions of the shape of the clusters or the consistency of the algorithms, and are not based on the underlying distribution of the data. Clearly, there is no universal validation technique, as there is no universal clustering algorithm.

4.5. Examples of clustering-based analysis of microarray data

4.5.1. Genomic profiling

Johnson et al. [50] used a combination of statistical and clustering methodologies to define genomic profiles and predictive networks of biological activity during the early stages of the atherogenic response to benzo(a)pyrene (BaP), an environmental hydrocarbon that initiates oxidative stress in vascular smooth muscle cells. k-means, fuzzy c-means, and hierarchical clustering were applied to genes found to be statistically significant via ANOVA to identify genes modulated by atherogenic insult in a redox-sensitive manner. Of interest was the finding that many of the genes identified as redox-regulated targets via ANOVA test, clustered together using clustering methodologies. The three nonsupervised methods resolved similar groups of genes and identified clones that were highly up-regulated by prooxidant alone, unaffected by antioxidant pretreatment, and neutralized by combined chemical treatments.

Hierarchical clustering was chosen to further resolve clusters of redox-regulated genes because the other methods forced clusters of similar size regardless of biological outcome, and antioxidant modification of gene-expression across treatments. This analysis readily identified genes that were modified by BaP in the absence of antioxidant and neutralized in the presence of antioxidant. Clustering analysis was employed in this study because there were underlying patterns of gene-expression which were not readily discernable using classical statistical

methodologies. Clustering was found to perform well in segregating genes that were altered by redox status and finely separate different behavior within this class.

4.5.2. Subclassification of diffuse large B cell lymphoma

In another example, Alizadeh et al. [9] presented a study of classification of human cancers for three adult cancer types: diffuse large B-cell lymphoma (DLBCL), follicular lymphoma (FL), and lymphocyte leukemia (CLL). The goal of the study was to determine whether gene-expression profiling could subdivide cancer types into molecularly distinct diseases with more homogeneous clinical behaviors. Instead of limiting the analysis to these types, additional samples were obtained from normal lymphocyte subpopulations under varying activation conditions, and some lymphoma and leukemia cell lines. Expression ratios were used against a common reference for a total of 96 samples.

The first step was the creation of a hypothesis using hierarchical clustering and other techniques. For this, these investigators localized genes that differentiate lymphocytes (gene-expression signature) and then clustered by the algorithm (similar expression profiles). This process allowed the researchers to select groups of genes with similar expression up-regulated in specific tissues, and to generate hypothesis about the data, by visually observing the patterns of gene activity in each group.

It was hypothesized that there could be two subclasses of DLBCL cancer types that could be derived from distinct stages of normal B-cell differentiation, and could be detected by genes characteristics of *germinal centre B cells*. To investigate this hypothesis, hierarchical clustering was employed for the genes characteristics of germinal centre B cells, and splitting the data in two classes, calling them *GC* and *activated*, respectively. Based on this analysis, it was concluded that a distinct class of DLBCLs was derived from the germinal centre B cell and retained gene-expression programs, and presumably many of the phenotypic characteristics, of this stage of B-cell differentiation.

In conclusion, hierarchical clustering and gene profiling allowed subclassification of large B-cell lymphoma (DLBLC) into two groups, derived from different stages of B-cell differentiation and activation. Eventually, this kind of analysis could be carried out using supervised analysis, since the classes are known, and the gene-expression signatures could be defined without clustering. Thus, the clustering process and visualization helped to create relevant hypotheses directly verifiable by ocular inspection of the graphical images.

4.5.3. External validation of microarray analysis

Duan and Zhang [51] presented an example of external validation for clustering algorithms applied to microarray data. The goal of the study was to present an application of k-means with weights at a variable level to compensate for loss of cell cycle synchrony in time-series experiments. In cell-cycle experiments, samples of mRNA are extracted at specific time intervals. The mRNA samples are derived

from a pool of cells that is initially synchronized, but that loses synchronization over time. Loss of synchronization means that different cells in the pool may be at different stages of the cell cycle, and therefore expressing differently. The difference among cells within the same sample is quantified by the variance in measurements. Samples closest to the start of the experiment are given a higher weight than samples in later stages of the experiment. The weights are computed as a function of the time between the start of the experiment and the extraction time.

The algorithm selected, a weighted version of k-means, is expected to perform better than standard k-means. The authors compared the algorithms using an adjusted Rand index between the result of the clustering algorithm and the true partition for selected genes. As a test of the validity of the approach, the algorithms are applied to artificial data, where the true partitions are known, and the Rand index is computed for all genes. Subsequently, biological information is used to evaluate the algorithms applied to microarray data based on a listing of protein complexes that was retrieved from an existing database. The "true" partition of the genes is defined by sets containing genes whose products belong to the same complex. In both cases, the weighted k-means algorithm performed better than the standard k-means algorithm, but the real contribution of the approach was the specification of viable additional information for external validation in microarray analysis.

A critical consideration for this approach is that some assumptions must be made at the moment of selecting a "true" partition to validate the data. In this study, the assumption used was that subunits in a protein complex present a high level of coregulation that is partially observable in terms of mRNA expression [52]. The degree to which this assumption is true indicates the success of the technique when applied to "real data."

4.5.4. Discussion

The aforementioned clustering applications areas are representative of gene profiling, subclass discovery, and external validation. The most frequently used algorithm for practical applications is the hierarchical algorithm, with both Euclidean and correlation distances. The preference for this algorithm is likely based on its ability to show a hierarchy of groups, to be applied without knowledge of the number of clusters, and to sort the data based on similarities. The k-means clustering method is less frequently used in practical applications because its simplicity of use is overcome by its low performance in biological systems compared to other algorithms. Most surprising, however, is that model-based algorithms are seldom used, in spite of the fact that they have more statistical support than other algorithms.

Bibliography

[1] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[2] C. Fraley and A. E. Raftery, "MCLUST: software for model-based clustering and discriminant analysis," *J. Classification*, vol. 16, pp. 297–306, 1999.

[3] D. Fasulo, "An analysis of recent work on clustering algorithms," Tech. Rep. 01-03-02, Department of Computer Science and Engineering, University of Washington, Seattle, Wash, USA, 1999.
[4] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
[5] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, 1998.
[6] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *J. Comput. Biol.*, vol. 6, no. 3-4, pp. 281–297, 1999.
[7] A. Brazma and J. Vilo, "Gene expression data analysis," *FEBS Lett.*, vol. 480, no. 1, pp. 17–24, 2000.
[8] H. Chipman, T. Hastie, and R. Tibshirani, "Clustering microarray data," in *Statistical Analysis of Gene Expression Microarray Data*, Chapman & Hall/CRC, Boca Raton, Fla, 2003.
[9] A. A. Alizadeh, M. B. Eisen, R. E. Davis, et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.
[10] P. Tamayo, D. Slonim, J. Mesirov, et al., "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 6, pp. 2907–2912, 1999.
[11] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, no. 10, pp. 977–987, 2001.
[12] M. F. Ramoni, P. Sebastiani, and I. S. Kohane, "Cluster analysis of gene expression dynamics," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 14, pp. 9121–9126, 2002.
[13] F. Azuaje, "A cluster validity framework for genome expression data," *Bioinformatics*, vol. 18, no. 2, pp. 319–320, 2002.
[14] M. Ashburner, C. A. Ball, J. A. Blake, et al., "Gene ontology: tool for the unification of biology. The gene ontology consortium," *Nat. Genet.*, vol. 25, no. 1, pp. 25–29, 2000.
[15] T. Konishi, "Three-parameter lognormal distribution ubiquitously found in cdna microarray data and its application to parametric data treatment," *BMC Bioinformatics*, vol. 5, no. 1, pp. 5, 2004.
[16] J. Rougemont and P. Hingamp, "DNA microarray data and contextual analysis of correlation graphs," *BMC Bioinformatics*, vol. 4, no. 1, pp. 15, 2003.
[17] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
[18] L. Bullinger, K. Dohner, E. Bair, et al., "Use of gene-expression profiling to identify prognostic subclasses in adult acute myeloid leukemia," *N. Engl. J. Med.*, vol. 350, no. 16, pp. 1605–1616, 2004.
[19] M. L. Bittner, P. Meltzer, Y. Chen, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.
[20] I. Hedenfalk, D. Duggan, Y. Chen, et al., "Gene-expression profiles in hereditary breast cancer," *N. Engl. J. Med.*, vol. 344, no. 8, pp. 539–548, 2001.
[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2nd edition, 2001.
[22] J. Quackenbush, "Computational analysis of microarray data," *Nat. Rev. Genet.*, vol. 2, no. 6, pp. 418–427, 2001.
[23] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.
[24] E. Gose, R. Johnsonbaugh, and S. Jost, *Pattern Recognition and Image Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.
[25] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Orlando, Fla, USA, 1998.
[26] L. Bottou and Y. Bengio, "Convergence properties of the k-means algorithms," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds., pp. 585–592, MIT Press, Cambridge, Mass, USA, 1995.
[27] E. R. Dougherty, J. Barrera, M. Brun, et al., "Inference from clustering with application to gene-expression microarrays," *J. Comput. Biol.*, vol. 9, no. 1, pp. 105–126, 2002.

[28] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *J. Amer. Statist. Assoc.*, vol. 97, no. 458, pp. 611–631, 2002.
[29] S. Van Dongen, "A cluster algorithm for graphs," Tech. Rep. INS-R0010, National Research Institute for Mathematics and Computer Science, Amsterdam, The Netherlands, 2000.
[30] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity," *Inform. Process. Lett.*, vol. 76, no. 4-6, pp. 175–181, 2000.
[31] R. Sharan and R. Shamir, "CLICK: a clustering algorithm with applications to gene expression analysis," *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, vol. 8, pp. 307–316, 2000.
[32] J. Wang, J. Delabie, H. C. Aasheim, E. Smeland, and O. Myklebost, "Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study," *BMC Bioinformatics*, vol. 3, no. 1, pp. 36, 2002.
[33] A. V. Lukashin and R. Fuchs, "Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters," *Bioinformatics*, vol. 17, no. 5, pp. 405–414, 2001.
[34] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, USA, 1988.
[35] F. Azuaje and N. Bolshakova, "Clustering genome expression data: design and evaluation principles," in *A Practical Approach to Microarray Data Analysis*, D. Berrar, W. Dubitzky, and M. Granzow, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 2002.
[36] N. Bolshakova and F. Azuaje, "Cluster validation techniques for genome expression data," Tech. Rep. TCD-CS-2002-33, Department of Computer Science, Trinity College Dublin, Dublin, Ireland, September 2002.
[37] S. Guenter and H. Bunke, "Validation indices for graph clustering," in *Proc. 3rd IAPR- TC15 Workshop on Graph-based Representations in Pattern Recognition*, J.-M. Jolion, W. Kropatsch, and M. Vento, Eds., pp. 229–238, Ischia, Italy, May 2001.
[38] E. J. Salazar, A. C. Veléz, C. M. Parra, and O. Ortega, "A cluster validity index for comparing non-hierarchical clustering methods," in *Memorias Encuentro de Investigación sobre Tecnologias de Información Aplicadas a la Solución de Problemas (EITI2002)*, Medellín, Colombia, 2002.
[39] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Intelligent Information Systems Journal*, vol. 17, no. 2-3, pp. 107–145, 2001.
[40] Z. Lubovac, B. Olsson, P. Jonsson, K. Laurio, and M. L. Anderson, "Biological and statistical evaluation of clusterings of gene expression profiles," in *Proc. Mathematics and Computers in Biology and Chemistry (MCBC '01)*, C. E. D'Attellis, V. V. Kluev, and N. E. Mastorakis, Eds., pp. 149–155, Skiathos Island, Greece, September 2001.
[41] E. R. Dougherty and M. Brun, "A probabilistic theory of clustering," *Pattern Recognition*, vol. 37, no. 5, pp. 917–925, 2004.
[42] A. Kalton, P. Langley, K. Wagstaff, and J. Yoo, "Generalized clustering, supervised learning, and data assignment," in *Proc. 7th International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pp. 299–304, San Francisco, Calif, USA, August 2001.
[43] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo, "Validating clustering for gene expression data," *Bioinformatics*, vol. 17, no. 4, pp. 309–318, 2001.
[44] S. Datta and S. Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data," *Bioinformatics*, vol. 19, no. 4, pp. 459–466, 2003.
[45] M. K. Kerr and G. A. Churchill, "Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 16, pp. 8961–8965, 2001.
[46] L. Fisher and J. W. Van Ness, "Admissible clustering procedures," *Biometrika*, vol. 58, pp. 91–104, 1971.
[47] J. W. Van Ness, "Admissible clustering procedures," *Biometrika*, vol. 60, pp. 422–424, 1973.
[48] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.
[49] S. D. Kamvar, D. Klein, and C. D. Manning, "Interpreting and extending classical agglomerative clustering algorithms using a model-based approach," in *19th International Conference on Machine Learning (ICML '02)*, Sydney, Australia, July 2002.

[50] C. D. Johnson, Y. Balagurunathan, K. P. Lu, et al., "Genomic profiles and predictive biological networks in oxidant-induced atherogenesis," *Physiol. Genomics*, vol. 13, no. 3, pp. 263–275, 2003.
[51] F. Duan and H. Zhang, "Correcting the loss of cell-cycle synchrony in clustering analysis of microarray data using weights," *Bioinformatics*, vol. 20, no. 11, pp. 1766–1771, 2004.
[52] R. Jansen, D. Greenbaum, and M. Gerstein, "Relating whole-genome expression data with protein-protein interactions," *Genome. Res.*, vol. 12, no. 1, pp. 37–46, 2002.

Marcel Brun: Computational Biology Division, Translational Genomics Research Institute, 400 N. Fifth Street, Phoenix, AZ 85004, USA
Email: mbrun@tgen.org

Charles D. Johnson: Array Services, Ambion, Inc., 2130 Woodward, Austin, TX 78744-1837, USA
Email: cjohnson@ambion.com

Kenneth S. Ramos: Department of Biochemistry and Molecular Biology, University of Louisville, HSC-A Bldg, Room 606, 319 Abraham Flexner Way, Louisville, KY 40202, USA
Email: kenneth.ramos@louisville.edu

5 From biochips to laboratory-on-a-chip system

Lei Wang, Hongying Yin, and Jing Cheng

Biochip-based systems have enjoyed impressive advancement in the past decade. A variety of fabrication processes have been developed to accommodate the complicated requirements and materials for making such a device. Traditional microfabrication processes and other newly developed techniques such as plastic molding and microarraying are being explored for fabricating silicon, glass, or plastic chips with diverse analytical functions for use in basic research and clinical diagnostics. These chips have been utilized to facilitate the total integration of three classic steps involved in all biological analyses, that is, sample preparation, biochemical reaction, and result detection and analysis, and finally construct fully integrated smaller, more efficient bench-top or even handheld analyzers—laboratory-on-a-chip system. Meanwhile, biochip-based analytical systems have demonstrated diversified use such as the analyses of small chemical compounds, nucleic acids, amino acids, proteins, cells, and tissues. In this chapter, aspects related to biochips with different functionality and chip-based integrated systems will be reviewed.

5.1. Technologies for fabricating biochips

Depending on the materials used, micromachining technologies employed for fabricating the biochips can be very different. Photolithographic processing techniques are by far the most commonly used methods for producing microchannels in the surface of a planar silicon or glass substrate. One advantage of using these materials is that their electrophoretic and chromatographic properties and surface derivatization chemistries are extensively studied in many cases. Another advantage is that many established microfabrication processes could be easily modified and applied. Injection-molding, casting, imprinting, laser ablation, and stamping processes represent another category of fabrication methods for machining plastic substrate. The advantage for using plastic as substrate is twofold. One is that plastic is less expensive and easier to manipulate than glass or silicon-based substrates. Another advantage is the easiness in disposing it after use. The third category of methods for fabricating one type of the most widely used biochips, that is, microarrays, is robotic station-based microdispensing methods.

5.1.1. Photolithographic fabrication of silicon and glass

In the process of fabricating a glass-based microfluidic chip, a protective etch mask of high quality, the appropriate glass type, and the composition of etchant are equally important. Wet etching is the most widely used method for fabricating microfluidic channels and reservoirs in glass, silica, or silicon substrates. Among glass substrates, the soda lime microscope slide (catalog no. 12-550C, Fisher Scientific) was the one that has been used from time to time in the past due mainly to its high etch rates [1]. With that merit less aggressive etchants can be used along with the use of hard-baked photoresist to obtain the required shallow etches [2]. However, for deep etching of microchannels, a more resistant sacrificial etch mask has to be used. Typical examples include the use of Cr/Au and amorphous silicon as sacrificial etch masks. Among these etch masks, photoresist/Au/Cr etch mask can be etched in various types of HF-based etchants especially when HF/HNO3 is employed as etchants simply because HF/HNO3 attack other masks such as amorphous silicon. When Borofloat glass is used with the photoresist/Au/Cr etch mask, channels with 35 μm etch depth could be obtained in approximately 5 minutes in 49% HF. Another type of etch mask, amorphous silicon, can be deposited on the substrates through plasma-enhanced chemical vapor deposition [3]. Used as an etch mask, amorphous silicon proved itself possessing the best resistance to HF etching as well as the fewest defects. Channels with smooth sidewalls and with depth of 70 μm were achieved when amorphous silicon was utilized as the etch mask. Generally speaking, high-quality amorphous silicon can withstand twice as long as Cr/Au when used as etch masks, making it possible to etch twice as deep. Among all sorts of glass substrates investigated, Schott Borofloat glass was found to have the best etching quality and simplest processing [3]. Borofloat is a borosilicate glass produced using the float process. This type of glass is uniform in thickness and composition and has a smooth and flat surface requiring no mechanical polishing (Figure 5.1). Another advantage of this material is that its background fluorescence is several times lower than that of microscope slide, making it the ideal candidate for high-sensitivity experiments. To make a complete piece of microfluidic device such as chip-based capillary electrophoresis, a planar cover glass with holes connected to the lower electrophoresis channels has to be used to seal the channels. To drill the holes on the cover glass, a diamond-tipped drill bit is ideal for rapid drilling of individual holes (approximately 15 seconds for each hole), whereas a multitipped ultrasonic drill bit is suitable for the production of many chips with consistent hole patterns (approximately 15 minutes for each chip). Bonding of the etched and drilled glass of the same size and type can be done with three approaches. One is to sandwich the glass substrates with two polished graphite blocks and then place them in an evacuated furnace with a stainless steel weight on top [3]. Another method is to have a thin layer of silicon dioxide deposited on one side of the cover glass and then allow the top and bottom glass substrates to be anodically bonded. The yield of chips made this way can be much higher than that of the thermal fusing method. The third approach is to bond the quartz glass substrates with hydrofluoric acid. The advantages of this

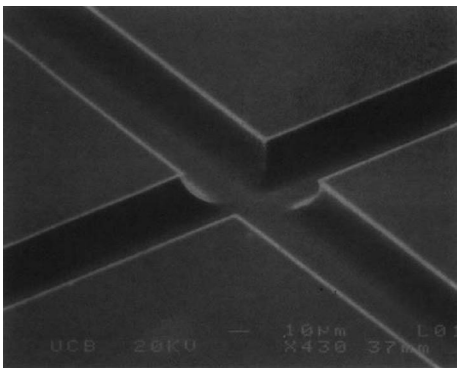


Figure 5.1. Scanning electron micrograph of the intersection of two 25 μm deep channels etched in Borofloat using a 1700 Å thick amorphous silicon etch mask. Striations on the side walls of the capillary are evident near the intersection. From [3], with permission.

method include low thermal damage, low residual stress (bonding at room temperature), and simplicity in operation [4]. Plasma etching of silicon master for hot embossing [5] and deep reactive ion etching of Pyrex using SF6 plasma were reported for fabricating microfluidic devices [6]. Microstereolithography, a method based on polymerization of photosensitive polymers by a focused UV beam, was used for the fabrication of 3D structures [7] and encapsulating material in microfluidic structures [8]. In a recent report, 3D microstructures were fabricated in a single exposure by a reduction photolithography using arrays of microlenses and gray scale masks [9]. Gray scale masks were also implemented in an excimer laser micromachining system to produce 3D structures with a continuous profile [10]. 3D-aligned microstructures have been fabricated by Tien et al., by pressing a multilevel Poly(dimethylsiloxane) (PDMS) stamp on a substrate and applying a soft lithographic technique every time a new level of the stamp came into contact with the substrate surface, to produce complex patterns of aligned microstructures [11].

5.1.2. Plastic microfabrication

The methods developed for fabricating microchannels and reservoirs in polymeric substrates include laser abalation, injection-molding, compression molding, casting, and X-ray lithography. Injection-molding and casting methods are in general called replication methods. Unlike laser etching method, the replication methods can generate capillary or reservoir with smooth surfaces, which are highly demanded by performing capillary electrophoresis.

Injection-molding of a capillary electrophoresis chip usually involves a multistep fabrication process. First, to obtain a "negative" image of the capillary channels, a silicon master has to be wet-etched through the standard photolithographic process. The fabricated silicon master has protruded features formed; the height

and width of these features corresponding to the specified dimensions of the capillary separation channel. Second, a "positive" metal mold is formed through electroplating against the silicon master. From this metal mother, daughters with the features same as that of the silicon master can be made, and these daughter metal molds may then be mounted on a mold insert. The insert can be used through the injection-molding or compression molding processes to produce hundreds of thousands of molded plastic chips using polymeric materials such as acrylic coploymer resin. The preparation of a sealed capillary electrophoresis chip was done by first drilling a few millimeters diameter through holes on the molded piece followed by thermal lamination of a thick sheet of Mylar coated with a thermally activated adhesive at raised temperature [12]. A microfluidic device with integrated spectrophotometric elements was injection-molded in poly(methyl methacrylate) (PMMA) recently, with the strategy combining the direct photopatterning, replica molding techniques, and a rapid cooled release mechanism in conjunction with material-material transfer [13].

Using casting process to fabricate polymeric chip is similar to the injection-molding process. The materials used including PDMS and PMMA [14, 15]. The polymer chip was cast against the wet-etched silicon master. Once formed, the PDMS replica can be peeled off easily. After punching the buffer reservoirs through, the cast chip can be placed on a slab of PDMS to form a sealed capillary electrophoresis device. In a report from Lee et al., microfluidic structures were transferred from quartz master templates to PMMA plates by using hot embossing methods. The relative standard deviation of the channel profile on the plastic chips was proved to be less than 1% [16]. Hot embossing was also used to fabricate microchannels on polycarbonate and Zoenor 1020 by other researchers [17, 18].

Apart from replication method, photoablation can also be used to machine plastic chips [19]. During the machining process, the laser pulses in the UV region can be transmitted through a mask to hit the selected areas on the plastic substrate. When the illuminated areas absorb the laser energy, the chemical bond within the long-chain polymer molecules are broken, and the photoablation generated debris such as gas, polymer molecules, and small particulate matter are ejected, leaving the desired channels and reservoir in the plastic chip. There is a variety of polymer materials that can be photoablated including polycarbonate [17], PMMA [16, 20], polystyrene, nitrocellulose, polyethylene terephthalate, and Teflon [21]. PMMA capillary electrophoresis chip has also been fabricated using soft X-ray lithography and a transfer or Kapton mask [22]. The main advantage of machining PMMA in soft X-rays is that narrow and deep channels (i.e., high aspect ratio) can be fabricated in the substrate. Several components were machined in PMMA including the injector, separation channel, and integrated fiber optic fluorescence detector [20].

5.1.3. Microarraying methods

The microarraying methods were developed for fabricating microarrays of genes, proteins, and other substances. These printing methods could be divided into

two types, that is, contact and noncontact printing methods. The contact printing method allows the microarray substrate to be physically contacted by the liquid dispensing mechanism. The arrayer uses metal pins of micron size or capillary tubing to deliver sample solution of nano- or even picoliters. The noncontact printing method allows the liquid dispensing mechanism to shoot the sample solution of nano- or even picoliters directly to the microarray substrate which is especially suitable for fabricating the protein or antibody microarrays where the maintenance of the protein structures is essential. The two most widely adopted dispensing approaches are piezoelectric-actuated glass capillary tips and solenoid valves [23]. Some representative developments in microarray technologies include BeadArray by Illumina, micromirror technology by NimbleGen, and thermal inkjet technology by Agilent. These methods are capable of producing readings far beyond the typical 40 000 spots per assay range, as well as typical application of gene expression profiling.

The BeadArray technology was first reported by David Walt and colleagues at Tufts University [24] and developed at Illumina as a platform for single nucleotide polymorphism (SNP) genotyping and other high-throughput assays. Each array is assembled on an optical imaging fiber bundle consisting of about 50 000 individual fibers fused together into a hexagonally packed matrix. The ends of the bundle are polished, and one end is etched to produce a well in each fiber. This process takes advantage of the intrinsic structure of the optical fibers in the bundle. After polishing and etching, a fiber bundle can hold up to 50 000 beads, each approximately 3 μm in diameter and spaced approximately 5 μm apart. This highly miniaturized array is about 1.4 mm across and has a packing density of 40 000 array elements per square millimeter—approximately 400 times the information density of a typical spotted microarray with 100 μm spacing. Each derivatized bead has several hundred thousand copies of a particular oligonucleotide covalently attached. Bead libraries are prepared by automated conjugation of oligonucleotides to silica beads, followed by quantitative pooling together of the individual bead types. The beads are stably associated with the wells under standard hybridization conditions. Basing on the BeadArray technology, the system, as implemented in a high-throughput genotyping service facility at Illumina, has a current capacity of one million SNP assays per day and is easily expandable [25].

Different from Illumina's BeadArray technology, NimbleGen builds its arrays using photo deposition chemistry with its proprietary maskless array synthesizer (MAS) system [26]. At the heart of the system is a digital micromirror device (DMD), similar to Texas Instruments' digital light processor (DLP), employing a solid-state array of miniature aluminum mirrors to pattern up to 786 000 individual pixels of light. The DMD creates virtual masks that replace the physical chromium masks used in traditional arrays. These virtual masks reflect the desired pattern of UV light with individually addressable aluminum mirrors controlled by the computer. The DMD controls the pattern of UV light on the microscope slide in the reaction chamber, which is coupled to the DNA synthesizer. The UV light deprotects the oligo strand, allowing the synthesis of the appropriate DNA molecule [27].

To fabricate high-density arrays, physical delivery techniques such as inkjet [28] or microjet deposition technology are used, too. Agilent's noncontact *in situ* synthesis process is an example known for its excellence. All the 60-mer length oligonucleotide probes, base-by-base, are printed from digital sequence files and the standard phosphoramidite chemistry used in the reactions allows for high coupling efficiencies to be maintained at each step in the synthesis of the full-length oligonucleotide. With Agilent's SurePrint technology, high-quality, consistent microarrays could be delivered to conduct in-depth gene expression studies [29]. The high level of multiplexing and modular, scalable automation of the above technologies shed a light on meeting the requirements for cost-effective, genome-wide linkage disequilibrium studies, which will open the door to personalized medicine.

5.2. Microfluidic control units

Microfluidic control is a critical to a lab-on-a-chip system. Fluid control tasks include acquisition and metering of both sample and reagents by the microchip from a specimen container or reservoir, speed and direction control for transporting sample and regents to different regions of the microchip for processing, and so forth. A range of micropumps and valves has been machined using microfabrication and microelectro mechanical system (MEMS) technology. Once integrated with other microchip-based devices these units may provide efficient fluidic control for the lab-on-a-chip systems.

5.2.1. Microvalves

Microvalve is an essential component for a lab-on-a-chip system. Appropriate utilization of microvalves will facilitate the storage of reagents, the priming of channels, the switching of liquid flow streams, and the isolation of specific areas of the chip during sensitive steps in the chemical processing to prevent leakage and pressure fluctuations.

Freeze-thaw valve. The freeze-thaw valve does not require any moving parts in the capillary channel and has no dead volume [30]. A small section of fluid inside the microchannel on a chip is made to act as its own shut-off valve upon freezing. The freezing process could be realized by using a fine jet of a mixture of liquid and gaseous carbon dioxide at approximately -65°C delivered from a cylinder of the compressed liquid. It has been demonstrated that localized freezing can stop the flow of fluid driven by the electroosmotic pumping. To make the cooling system compatible with the planar microstructures, a chip-based electrothermal cooling device such as a Peltier device may be applied.

Magnetic valve. Löchel et al. utilized a thin square-shaped membrane structure (2×2 mm) of electroplated NiFe alloy as the flow-controlling element for their magnetic valving system [31]. The magnitisable membrane structure was driven by the presence or absence of a magnet applied externally to the chip device. In the middle of the membrane, an integrated bar of the same ferromagnetic material amplifies the force for moving the membrane. The four edges of the membrane

were sealed against the silicon substrate. The magnetic valve is normally open and flow occurs if a high pressure is applied from the upper side of the valve. The application of a magnetic field drives the ferromagnetic membrane toward the valve seat and closes the valve. In another report, a more practical magnetic pump with similar principal has been made using both bulk micromachining and wafer bonding techniques [32]. This valve by default is normally closed when there is no excitation, which enhances safety if a valve failure occurs. When the required voltage is applied to the inductor mounted on a glass wafer, the magnetic fluxes will be generated and the permalloy/magnetic membrane will be become attracted to the upper electromagnet, and thus leave the valve open to allow the fluid flow through the valve seat. For the current of approximately 700 mA, the flow rate can go up to 600 μL/min. A new type of magnetic microvalve has been fabricated by Hartshorne et al. [33]. Flow control was achieved by moving a ferrofluid in a microchannel to open or close an intersection through an externally applied magnet.

Flow switch. A flow switch system has been developed by Blankenstein and Larsen and used as a valveless five-way valve [34]. This system consists of two syringe infusion pumps generating a constant flow rate and a microfabricated flow chip with 3 inlets and 5 outlets. The basic principle of this type of valve is as follows. A sample containing flow stream is centered and guided by two buffers on each side through a linear microchannel and leaves the flow chip via the middle outlet. Hence, if the flow ratio between the two control buffers is altered, the sample stream becomes deflected and is forced to enter one of the four outlet channels, depending on the set flow ratio. The time the sample flow stream is forced into the selected outlet is determined by the actuation/switching time and the volumetric flow rate inside the microchannel is controlled by precision driven syringe pumps. A microdevice called "flowFET," with functionality comparable to that of a field-effect transistor (FET) in microelectronics, has been developed [35]. The magnitude and direction of the electroosmotic flow (EOF) inside the microfabricated fluid channel can be controlled by a perpendicular electric field. In another study, a novel elastomeric microfluidic switch has been developed by Ismagilov et al. [36]. Two fluid switching methods were established by controlling the lateral position of a stream in multiphase laminar flow and the channel aspect ratio in tangentially connected microfluidic systems. Furthermore, an actuator based on a thermoresponsive hydrogel, that shrinks or swells with fluid from a separate reservoir and thereby displaces a PDMS membrane to actuate fluid in the microchannel underneath was reported lately [37]. Monolithic membrane valves, fabricated by sandwiching an elastomer membrane between etched glass fluidic channels, that are suitable for large-scale integration were presented [38].

5.2.2. Micropumps

Electrohydrodynamic pump. The electrohydrodynamic pump consists of two planar-processed, conductive electrodes. When the electrodes are in contact with fluids inside the microchannels, the pressure can be generated by ion dragging of fluids [39]. Applied voltage ranging from 300–500 V should in general result in

pressures on the order of inches of water. The dominant force in electrohydrodynamic pumping is the coulomb interaction with a space-charge region that is formed by injected or induced charges in the fluid. These charges are due to electrochemical reactions at the electrodes. Pumps like electrohydrodynamic pump are particularly suitable for application where many different fluid samples need to be transported from one place to another in a micromachined device.

Magnetohydrodynamic pump. An AC magnetohydrodynamic (MHD) pumping system has been presented by Lemoff and Lee, in which the Lorentz force is used to propel an electrolytic solution along a microchannel etched in silicon [40]. The micropump has no moving parts, produces a continuous (not pulsatile) flow, and is compatible with solutions containing biological specimens.

Electroosmotic pump. The first study on fluid flow driven by electroosmotic pumping in a network of intersecting capillaries integrated on a glass chip was reported in 1994 [41]. Controlling the surface chemistry and the potentials applied to the microchannels allowed accurate transport of sample and reagents of fixed volume from different streams to an intersection of capillaries.

Traveling-wave pumping. Directed movement of fluid and particles suspended in a solution can be achieved via traveling-wave pumping [42]. The driving force is generated from the applied four high-frequency square-wave voltages, with sequential phase differences of 90° to the micrometer-sized electrodes arranged in parallel and one next to another. The high-frequency traveling-wave field is able to drive the liquid forward but simultaneously may also trap microparticles present in the fluid on to the electrode edges through dielectrophoresis. The latter feature of traveling-wave pumping may be especially useful for "filtering" particles such as bacteria from a water sample.

Thermal capillary pump. The thermal capillary pump works by selectively allowing the DC current to flow through the addressed electrodes built inside the microchannel fabricated on the silicon chip. Nanoliter-sized discrete drops of fluid can be moved around through addressable local heating [43]. The electrodes were made by first depositing a 0.35 μm thick layer of aluminum on the silicon wafer using an electron beam coating technique, and then covering the aluminum electrodes sequentially with 1 μm SiOx, 0.25 μm SixNy, and 1 μm SiOx using plasma-enhanced chemical vapor deposition. This pump can accurately mix, measure, and divide drops by simple electronic control thereby providing a versatile pumping method with multiple functions.

Piezoelectric pump. One of the earliest piezoelectric pumps was built in 1993 with two glass plates and a silicon wafer [44]. A pressure chamber and a raised flat surface suspended with a thin diaphragm are formed on the upper glass plate. The piezoelectric actuator is placed on the raised flat surface. In order to guide the flow of the pumped liquid, two check valves made of polysilicon are fabricated on the silicon wafer at the inlet and outlet of the pressure chamber. When the piezoelectric actuator is switched on through the applied periodic voltages, the liquid is driven to the outlet. When the actuator is switched off, the liquid flows from the inlet

into the pressure chamber. A further development has been a dynamic passive valve whose performance is superior to that of the traditional static passive valves. To stop the flow of the fluid in a static passive valve, a mechanical element such as a flap, a sphere, or a membrane is usually used. In contrast, the dynamic passive valve uses flow-channels having a simple truncated pyramidal shape [45, 46]. Improvement has been also made in building a miniature piezoelectric pump with high bubble tolerance and self-priming capability [47]. The pumprate is about 1 mL/min for water and 3.8 mL/min for gas. The driving electronics have a volume of $30 \times 13.5 \times 8\,\text{mm}^3$ which allows the pump to be suitable for use in any portable lab-on-a-chip system. Furthermore, using injection-molding method a piezoelectric pump was produced with self-priming capability and a high pumprate up to 2 mL/min for liquid and 4 mL/min for gases [48].

Magnetic pump. An electromagnetically driven peristaltic micropump on a silicon wafer has been fabricated [32]. This pump can be operated at a maximum flow rate of 30 μL/min at 10 Hz in peristaltic motions with a DC current of 300 mA. It allows for bidirection pumping flows. In a recent report, a novel microperistaltic pump with two separate parts is presented. One part of the pump was integrated into a small disposable cartridge and the other was made reusable as an integrated part in the analytical device. Regarding the first part, three identical chambers were fabricated in the cartridge and actuated in peristaltic mode by strong permanent magnetic forces as well as restoring forces. The peristaltic timing was generated by the second part, which is a reusable rotating permanent sector magnet. A maximal flow rate of 3.1 mL/min and a backpressure of 20 kPa were obtained with this pump [49].

Other type of pumps. A thermally driven phase-change nonmechanical micropump has been investigated theoretically and experimentally [50]. The pumping of fluids was realized by using the actuation of a moving vapor slug (bubble) generated by suitably phased heating elements along the channel. Pumping of aqueous and organic liquids in millimeter- and micrometer-sized channels by controlling both spatially and temporally the concentration of redox-active surfactants using an electrode array was demonstrated by Gallardo et al. [51]. Surfactant species generated at one electrode and consumed at another were used to manipulate the magnitude and direction of spatial gradients in surface tension and guide droplets of organic liquids through simple fluidic networks. Prins et al. have demonstrated controlled fluid motion in 3D structures with thousands of channels using the electrocapillary pressure by electrostatically controlling the channel-fluid interfacial tension [52]. The velocities of several centimeters per second are nearly two orders of magnitude higher than the velocities demonstrated by other electrofluidic actuation principles.

5.3. Sample processing

Sample preparation is the first stage of a lab-on-a-chip system. It generally implies the ability to process crude biological samples such as blood, urine, water, and so

forth to isolate target molecules or bioparticles of interest such as nucleic acids, proteins, or cells. Currently, most of the analytical methods used in biomedical research and clinical applications analyze samples at volumes greater than 2 μL. Handling and processing of microsamples (e.g., μL and sub-μL volume) is difficult. Analysis of sub-μL volumes of sample has known problems such as loss of sample on the walls of pipette tips, loss by evaporation, loss of the targeted analyte because of adsorption onto the tubing walls or containment vessels during manipulation and processing period, and difficulty in obtaining a representative sample from a nonhomogeneous specimen. Additionally, the low concentration of the analyte may restrict the scale of miniaturization. In many cases, the analytes are usually present at extremely low concentration, for example, 100 molecules/mL. Hence, in a 1 μL sample there is less than one molecule of the analyte, and thus this degree of miniaturization is impracticable. Sample miniaturization is suitable for molecular analysis of genomic targets. Generally speaking, there are approximately 4400–11 000 white cells in a 1 μL of adult human blood. In theory the DNA molecules from a single white cell are sufficient to allow the amplification of the region of interest millions of times through the use of molecular technologies such as PCR. For a blood specimen, suppose the white blood cell count is 10 000/μL, the average volume for a sample to contain one white blood cell is 100 pL. In the event of detecting rare cell types or microorganisms (e.g., detection of cancerous cells, fetal cells in maternal circulation, assessment of minimal residual disease), insisting on the use of reduced volume of samples is no longer practical. Under these circumstances, sample sizes compatible with detection will have to be determined by the expected cell frequency or microbial load and sample volumes ranging from 100 μL to 5 mL may be desired. Moreover, specific selection (e.g., dielectrophoresis technology) or a preconcentration step has to be adapted to ensure the presence of the desired cells or microorganisms.

5.3.1. Microfiltration

To analyze nucleic acid by a lab-on-a-chip system the nucleic acids released from white blood cells usually have to be amplified by various amplification technologies such as PCR or strand displacement amplification (SDA). However, these amplification processes might be inhibited by hemoglobin released from red blood cells. Hence, a fundamental consideration in designing the microfilter chips for sample preparation is to facilitate the largest possible isolation of white blood cell populations or nucleic acids with very low red cell or hemoglobin contamination. All microfilter chips so far have been fabricated directly from silicon using both conventional wet etching and reactive ion etching. Different structural designs were explored, including simple arrays of posts [53], tortuous channels, comb-shape filter, and weir-type filters [54]. The general structure of a microfiltration chip is an etched chamber that contains the filter element across the entire width of the chamber. The structure is capped with a planar glass. Sample is normally pumped into the microfilter chip. According to the design, different particulate components should be trapped either at the front entrance or within the filter bed.

The study of microfilter-facilitated cell separation soon revealed that the deformability of cells plays a critical role in the separation efficiency. The filter dimensions were initially designed according to the reported sizes of blood cells obtained from morphological measurements of the stained cells. Yet, filtration of white and red blood cells was found to be influenced by the cell concentration, applied pressure, viscosity of the medium, and the size of the filter port. And it was discovered that red blood cells with relatively stable discoid architecture readily align themselves to facilitate passage through a 3 μm gap while highly deformable white blood cells with spherical diameters in excess of 15 μm will pass through filter gaps of only 7 μm. Thus, optimization of filter geometry was performed and weir-type filters with a filter gap of approximately 3 μm were proved to be effective in isolating large-sized white blood cells with relatively high yield [54]. For genomic studies using DNA/RNA amplifications, it is not essential to achieve high efficiency in white cell collection, but rather to achieve an adequate number of cells for successful amplification. Thus, a filter system that receives 1.0 μL of whole blood containing approximately 5000 white blood cells would be effective if the resulting white cell collection was only 10% (i.e., collected 500 cells) provided that the contained red cell population was less than 50 000 cells. Therefore, a system that isolates white blood cells with 10% efficiency and removes red blood cells with 99% efficiency will meet requirements. For the isolation of cells with very small sizes (e.g., bacterial or viral particles) or specific types or subtypes (e.g., CD4+), microfilter chips may be incompetent despite their effectiveness in removing red blood cells from blood. The following two isolation approaches may be found useful.

5.3.2. Magnetic cell sorting

A microfluidic structure has been made in silicon to enable the magnetic cell sorting [55]. An enrichment rate of more than 300-fold has been achieved. However, it was impossible to control the interaction time of particles with magnet due to the parabolic flow profile in microchannel. In addition, build-up of magnetic particles increased the magnetic field gradient inside the channel and consequently entrapment of particles was observed.

5.3.3. Electronic cell separation

Spiral gold electrodes were fabricated on the glass substrate. The electrode array consists of four parallel spiral electrode elements energized with phase-quadrature signals of frequencies between 100 Hz and 100 MHz. Depending on the frequency and phase sequence of applied voltages, the 3D forces generated by spiral electrodes could result in cell radial motion, levitation, and trapping. The chip bearing spiral electrodes has been used for the enrichment of breast cancer cells and isolation of breast cancer cells from blood [56]. Complicated design of platinum/titanium and indium tin oxide electrodes have been fabricated also on glass substrate for cell manipulation [57]. Negative dielectrophoresis has been employed in this case for concentrating and switching the particles at flow speed up to 10 mm/s.

In addition, planar microelectrodes were used to trap viral particles when a phase-shifted high-frequency AC signal is applied [58]. Moreover, individually addressable microelectrode array fabricated on silicon substrate has been used for the isolation of cultured cervical carcinoma cells from human blood [59]. This demonstrated the possibility of further integrating cell isolation devices with other microdevices through the use of established silicon processing technologies. Recently, a novel method for continuous cell separation was developed by integrating traveling-wave dielectrophoresis and laminar flow on a single biochip-based device, and the separation of Jurkat cells from red blood cells has been achieved [60].

5.4. Biochemical reaction

Biochemical reaction may include various types of chemical or enzymatic reactions such as chemical labeling, DNA amplification using PCR or SDA, or DNA restriction enzyme digestion.

5.4.1. Amplification of nucleic acids

The amplification of nucleic acids has been performed in microchips fabricated from different substrates such as glass [61, 62, 63], silicon-glass [59, 64, 65, 66], and plastics [67, 68]. Both thermal [69, 70] and isothermal amplification techniques were demonstrated [64, 71]. The reaction volumes varied from a 1 μL [68] togreater than 25 μL [72]. The silicon-glass microchips were bonded by using either silicone rubber [65] or anodic bonding [70]. The size of the amplification products ranges from approximately 50–1600 bp. Thermal cycling was achieved either by an on-chip polysilicon thin film heater or externally by means of a Peltier heater-cooler, or an infrared irradiation [66, 72]. Nucleic acids have been amplified in these microchips using conventional hot-start PCR, LCR, DOP-PCR [69, 72, 73], multiplex PCR, and SDA [64, 73]. RNA has been amplified using the single-step RT-PCR protocol [74]. Rapid PCR was achieved recently on a microchip-based PCR device using flexible printed circuit technology. An new digital temperature control system was developed by introducing a heater/sensor switching procedure. Temperature stability within $\pm 0.3^{\circ}$C and a transitional rate of 8°C/s during heating/cooling was achieved [75].

Surface chemistry plays a significant role in microchip amplification reactions [70]. Various passivation procedures have been tested and several were identified as PCR and LCR friendly. Covering a silicon surface with a thermally induced silicon dioxide layer (thickness of 2000 Å) is the most effective passivation procedure discovered so far for nucleic acid amplification reactions [72]. Isothermal nucleic acid amplification techniques (e.g., nucleic acid sequence-based amplification and SDA) are candidate techniques for a microchip format. These techniques do not require the use of the heater-cooler system and therefore greatly simplify the construction and operation of a microchip for nucleic acid analysis and should prove energy saving.

5.4.2. Other chemical reactions

Apart from the DNA/RNA amplification performed in various microchips, other chemical reactions have also been investigated using microchips. For example, both quartz and glass microchips have been fabricated for performing capillary electrophoresis and postcolumn reaction [76]. On-chip postcolumn reaction of *o*-phthaldialdehyde and amino acids generated theoretical plate numbers up to 83 000 and approximately 90 milliseconds peak widths. Approximately 10% degradation efficiency was due to the reactor geometry. Apart from that, it was found through the study that pH differences in the mixing solutions play a role in the efficiency of the postcolumn reactions. In another report, enzymatic reactions were performed within a microfabricated channel network [77]. Precise concentrations of substrate, enzyme, and inhibitor were mixed in nanoliter volumes using electrokinetic flow. Reagent dilution and mixing were controlled by regulating the applied potential at the terminus of each channel, using voltages derived from an equivalent circuit model of the microchip. The β-galactosidase-catalyzed hydrolysis of resorufin β-D-galactopyranoside was used as a model system for enzyme kinetic and inhibition determinations. The microchip approach assay allowed the studies to be completed with significant time-savings and reduction of reagent consumption by more than 4 orders of magnitude while delivering results consistent with conventional approaches.

5.5. Result detection

Result detection may be facilitated by microchannel-based separation approach, microarray-based affinity binding approach, and so forth.

5.5.1. Microchannel-based separation methods

One distinct advantage for microfabricated chips is that they can be utilized as platforms for multipurpose liquid sample handling and analysis. As a result, a variety of separation methods have been developed for use with microchips. The methods implemented on chips include free-solution capillary electrophoresis, capillary gel electrophoresis, micellar electrokinetic chromatography, isotachophoresis, isoelectric focusing, open-channel electrochromatography, and free-flow electrophoresis.

Free-solution capillary electrophoresis. Free-solution capillary electrophoresis was the earliest capillary electrophoresis transferred into microchip manifold. Representative pioneer report was jointly made by Harrison's and Manz's groups [78]. Using glass capillary electrophoresis chip, they performed the free-solution capillary electrophoretic separation of 6 γ-fluorescein isothiocyanate-labeled amino acids in approximately 15 seconds. Many works have been done since then. In a study conducted by Ramsey's group [79], free-solution capillary electrophoresis has been performed on a microchip capillary electrophoresis device. It is the first

time that single chromophore molecules were separated and then counted using confocal microscopy. Another study conducted by the same group demonstrated the separation of rhodamine B and dichlorofluorescein in 0.8 milliseconds [80]. In this study a separation channel of 200 μm long and 26 μm wide was fabricated and used to achieve a 100-fold decrease in electrophoretic separation. This high-speed microchip electrophoresis has also shown improved efficiency and this was made possible mainly by reducing the injected sample plug width and joule heating to increase the plate height. This system undoubtedly will become a valuable measurement for monitoring millisecond time-scale kinetics for chemical and biochemical reactions required commonly in ultra-high-throughput drug screening processes.

Capillary gel electrophoresis. Gel electrophoresis is the most common approach for separating biopolymers such as nucleic acids and proteins. Due to its wide application and huge commercial value, gel electrophoresis has been exploited for quite a few years in microchip platform. Effenhauser et al. [81] were the pioneers in this transferring process. They reported the first case in 1994 where noncross-linked 10% T polyacrylamide-filled microchannels machined on planar glass substrates were employed for size separation of phosphorothioate oligonucleotides ranging from 10–25 bases. The separation was obtained in 45 seconds with a plate height of 200 nm. Since then, a lot of research activities have been undertaken to advance this technique and broaden its applications. More review on this technique is presented in a later section.

Micellar electrokinetic chromatography. Micellar electrokinetic chromatography (MEKC) was initially developed by Terabe et al. [82]. In MEKC, surfactants with concentrations larger than their critical micelle points are added to the separation buffer facilitating the separation of uncharged solutes based upon differential partitioning. The separation of charged molecules is determined by electrophoresis, electrostatic interactions, solute complexation with the surfactant, and also partitioning between two phases. Microchip-based MEKC was first reported by Moore et al. to separate three coumarin dyes in a separation buffer of 50 mM SDS and 10% methanol [83]. At field strength lower than 400 V/cm, excellent reproducibility was demonstrated. In another study, the separation of 8 most common biogenic amines was achieved in 75 seconds by running the MEKC in a glass chip and also the biogenic amines from soy sauce samples were identified by the same approach [84]. A quite different separation format was reported by von Heeren et al., where a glass capillary electrophoresis chip with cyclic channels was fabricated for achieving separation of 6 fluorescein isothiocyanate-labeled amino acids in a few seconds [85]. The MEKC separation efficiency in this study was found comparable to that obtained by chip-based gel electrophoresis. Chip-based MEKC has shown increased separation efficiency and several 10-fold declines in separation time, when compared to conventional MEKC performed in fused-silica capillary. Additionally, the efficient heat dissipation in glass, silica, or silicon chips enables the application of very high field (up to 2 kV/cm) for separation achievable in millisecond to second time.

Isotachophoresis. The transfer of isotachophoresis on a glass-glass microchip was achieved to separate two herbicides [86]. The separated herbicides paraquat and diquat with concentrations as low as 2.3×10^{-7} M were detected by Raman spectroscopy detection. The Raman microprobe was directly coupled to the microchip without any interfacing. The Raman spectra were generated by using a 532 nm NdY-VO4 laser of 2 watts and collected at 8 cm^{-1} resolutions with a holographic transmissive spectrography and a cooled charge coupled device.

Isoelectric focusing. The capillary isoelectric focusing was performed in the separation channel with a width of 200 μm, depth of 10 μm, and length of 7 cm etched on a glass microchip [87]. The researchers found that compared to the chemical and hydrodynamic driven mobilization, EOF driven mobilization (occurring simultaneously with focusing) is most suitable for use with chip format due to its high speed, EOF compatibility, and low instrument requirements. Using chip-based capillary isoelectric focusing a mixture of Cy5-labeled peptides could be focused in less than 30 seconds with plate heights of 0.4 μm.

Open-channel electrochromatography. Other than capillary electrophoresis, microchip capillary platform has been adapted for a chromatographic technique called open-channel electrochromatography (OCEC) [88, 89]. In this circumstance, electroosmotic pumping was used to move the mobile phase through a serpentine-shaped microchannel. In the first study reported by Jacboson et al., the interface of the microchannel was chemically modified with octadecylsilane as stationary phase [88]. The separation of three neutral coumarin dyes was demonstrated in a mobile phase containing 10 mM sodium tetraborate and 25% acetonitrile. In a further study, solvent programming has improved the efficiency for OCEC [89]. The computer-controlled application of voltages was applied to the terminals of the chip for adjusting the isocratic and gradient elution condition. The researchers found that linear gradients with different slopes, start times, duration times, and start percentages of organic modifier are important elements for enhanced selectivity and reduced assay time. A complete run including fast reconditioning took only 60 seconds to accomplish.

Free-flow electrophoresis. In the separation of biopolymers or cells 2D methods are inherently more powerful than 1D method since the resulting peak capacity is increased. Free-flow electrophoresis (FFE) is one of these methods. Different from other 2D separation methods such as 2D gel electrophoresis where a supporting medium (like gel), high salt concentrations or even organic solvents have been used, FFE is very gentle in the separation conditions used and therefore especially suitable for separations involving cells and proteins. For example, Raymond et al. have performed FFE initially in a micromachined silicon device for continuous separation of rhodamine-B isothiocyanate-labeled amino acids in 20 minutes with an applied voltage of 40 v [90]. Later, they further separated high molecular weight compounds, such as human serum albumin, bradykinin, and ribonuclease A, in a 25-μL volume FFE microstructure [91]. Also continuous separation of more complicated samples such as tryptic digests of mellitin and cytochrome

c were obtained. The concentration of samples was detected using laser-induced fluorescence detection. In another report dielectrophoretic free-flow fractionation was conducted to separate human leukemia (HL-60) cells from peripheral blood mononuclear cells in a planar glass substrate plated with gold interdigitated electrodes [92]. With this technique an appropriate AC voltage signal was applied to the microelectrodes, the cells were levitated and suspended in the separation chamber with different equilibrium heights due to the dielectrophoretic forces generated by the applied AC signal. The separation of cells was eventually achieved based on the balance of dielectrophoretic, gravitational, and hydrodynamic lift forces they were subjected to. Two reports published at the same time described a method very similar in principle to that of FFE, differing in that microfabricated fractal arrays were adapted to replace the free zones created in FFE for achieving much higher efficiency and separation resolution [93, 94]. Although both works, were at their early stages the calculated separation of DNA with size ranging from 100–20 000 bp were predicted. Once realized, this technique may facilitate simple set-up and automation, and also the device cost might be less than 1 dollar and so it could be disposed after single use.

Pulsed-field electrophoresis. In a study by Austin's group, pulsed-field electrophoresis (PFE) was transferred into the microfabricated silicon devices where arrays etched on the silicon were utilized as separation matrix rather than gels used in conventional PFE [95]. The study indicates that the motions of biopolymers in the microarray matrix are more uniform compared to what happened in gels. Less dispersion in displacement is therefore anticipated which may lead to reduced band broadening and improved resolution. The separation can be greatly increased as megabase DNA will not be trapped in the array—arresting of long molecules in a gel will not happen here, so much higher separation voltage can be applied. Excellent heat dissipation through the silicon substrate further supported the application of higher fields.

Nucleic acid analyses. The analyses of nucleic acids can be divided into two main categories. One is the fragment sizing in most cases related to the detection of DNA mutations. The other one is DNA sequence analysis. For the first category polymer solution gel capillary electrophoresis has been used as the main separation media. One of the earliest separation cases, the rapid sizing of PCR amplified HLA-DQα alleles as well as the spiked DNA marker with size ranging from 72–1353 bp was obtained in approximately 2 minutes in a glass capillary electrophoresis device [96]. Hydroxyethyl cellulose (HEC) was used to form the entangled free-solution sieving matrix in this study. Apart from glass chips, plastic chips have been fabricated and used for fragment sizing [15, 97] and detection of single DNA molecules [15]. Using injection-molded acrylic capillary electrophoresis chip and HEC as sieving matrix all fragments in the DNA marker with size ranging from 72–1353 bp was baseline resolved in 2.5 minutes. The standard deviation for run-to-run is less than 1% and for chip-to-chip is between 2%–3% [97]. Also PDMS molded capillary electrophoresis chip has been used together with hydroxypropyl cellulose as sieving media in the separation of the marker same as that used in reference 6 and

36. In the study of single DNA molecule detection efficiency larger than 50% was obtained using the same device [15]. Fused-silica capillary electrophoresis chip has been fabricated and used for fast DNA profiling [97]. In this case a replaceable denaturing polyacrylamide matrix was employed for baseline-resolved separation of single-locus short tandem repeats amplicons. The complete separation of four amplicons containing loci of *CSF1PO*, *TPOX*, *THO1*, and *vWA* was achieved in less than 2 minutes representing a 10- to 100-fold increase in speed compared to the conventional capillary or slab gel electrophoresis systems. In another report, glass capillary array electrophoresis chip filled with HEC solution was used for high-speed DNA genotyping [98]. Twelve DNA samples with the largest fragment size of 622 bp were separated in parallel in less than 3 minutes (Figure 5.2). For the detection of all lanes a laser-excited confocal fluorescence scanner was developed for achieving a temporal resolution of 0.3 seconds. In a study reported by Ogura et al., ribosome RNA samples were separated in an injection-molded plastic microchannel with a cross-section of $100 \times 40\,\mu$m and an effective length of 1 cm [99]. The sieving matrix employed is hydroxypropylmethylcelulose and the detection of RNA, less than what can be obtained from a single cell, is achieved using a fluorescent microscope equipped with a photometer. Recently, a PDMS chip-based temperature gradient capillary electrophoresis was developed for fast screening of SNPs. A temporal temperature gradient with a precision of 0.1°C per step was applied on the chip during the separation. The homoduplexes and heteroduplexes were baseline resolved [100].

Ultra high speed DNA sequence analysis has been achieved on a glass capillary electrophoresis chip where a denaturing 9% T and 0% C polyacrylamide solution was used as the separation media [101]. When a four-color detection scheme was used the readout of approximately 200 bases was obtained in 10 minutes in an effective separation length of 3.5 cm. After optimization of both electrophoretic channel design and methods, much higher readout in four-color DNA sequencing was obtained, that is, 500 bases in less than 20 minutes (Figure 5.3) [102]. For the purpose of fast DNA sequence analysis, when a 96-channel array chip is used one can then easily see how significant the production rate could be compared to the conventional DNA sequencers. A fast sequencing system named BioMEMS-768 is being commercialized by Network Biosystems. Specific features of the system include throughput of 5 Mbp per day with a read length of 800 bp or greater and operating cost reduction of at least 50% compared to current capillary-based systems [103].

Immunoassay. One of the main uses of microchip capillary electrophoresis is for immunoassay where sensitivity and specificity of antibody-antigen interaction is critical. The ability to separate and quantify immunological reactants and products on-chip has been demonstrated [85, 104, 105]. In a clinically related study microfabricated fused-silica chip has been made for the separation and quantitation of free and bound labeled antigen in a competitive assay [105]. The microchip-based capillary electrophoresis analysis could detect cortisol present in blood serum over the range of clinical interest (1–60 μg/dL) without any sample pretreatment.

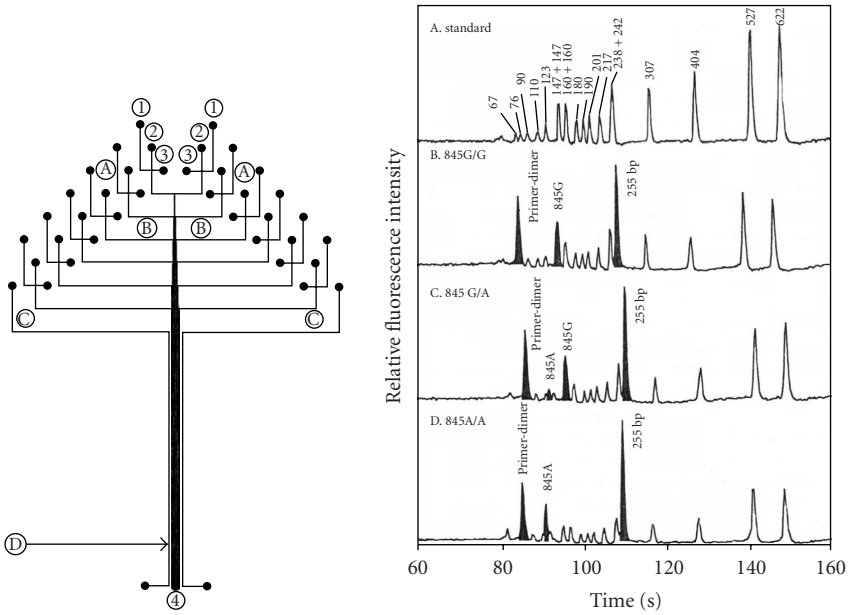


Figure 5.2. (a) Mask design used to photolithographically pattern the 12-channel CAE chips. (A) Injection channels, 8 mm long. (B) Separation channels, 60 mm long. (C) Optical alignment channels. (D) Detection region, ∼45 mm from the junctions of injection and separation channels. (1) Injection waste reservoirs. (2) Ground reservoirs. (3) Injection reservoirs. (4) High-voltage anode reservoir. The actual chip size was 50 mm × 75 mm. (b) Representative electropherograms of the three different HLA-H nucleotide 845 genotypes (B–D) generated from the microfabricated capillary array electrophoresis chip, along with the pBR322 *Mspl* standard ladder (A). The HLA-H peaks are shaded. From [98], with permission.

The separation and detection needs only 30 seconds to accomplish. In another report micellar electrokinetic capillary chromatography has been performed on a glass microchip fabricated with cyclic planar structure [85]. A competitive assay for theophylline (an asthma treatment drug) presented in serum has been conducted. The adsorption of proteins onto the uncoated walls of the injection channel can be overcome by adding some sodium dodecyl sulfate-containing buffer to the reaction mixture before injection. The separation speed is approximately 50 times faster than the conventional capillary electrophoresis analysis. Free-solution analysis of serum theophylline has also been performed on a capillary electrophoresis chip [105].

5.5.2. Microarray-based affinity binding assay

Analyses of nucleic acids. A variety of DNA chips have been prepared for DNA mutation detection [106, 107], SNP analysis [108, 109], DNA resequencing [110],

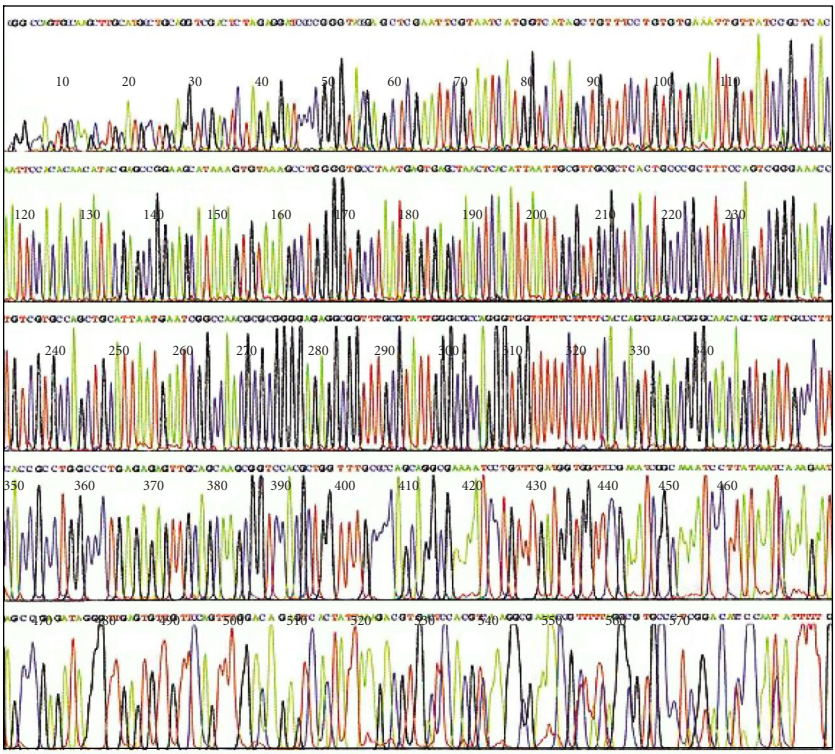


Figure 5.3. Analyzed four-color M13 DNA sequencing traces from a CE chip. Separation was performed on a 7 cm-long channel with a 100 μm twin-T injector using 4% LPA as the separation medium at 40°C. Separation was performed with a voltage of 160 V/cm, and the detector was 6.5 cm from the injector. Only 0.2 μg of DNA template was employed per reaction, and 1 μL of the final reaction solution (33%) was loaded on the chip. This run was complete in less than 20 minutes. From [102], with permission.

and gene expression studies [111, 112]. To screen for a wide range of heterozygous mutations in the 3.45 kilobases exon 11 of the hereditary breast and ovarian cancer gene *BRCA1*, a glass-based DNA chip with a density of 96 600 sites was used [106]. The oligonucleotide probes (each with a length of 20 nucleotides) were synthesized in situ using a light-directed synthesis method. Each assay requires more than 4 hours to complete. Comparatively speaking, performing DNA mutation detection using microfabricated silicon bioelectronic chip has obvious advantages in terms of saving time. Discrimination among oligonucleotide hybrids with widely varying binding strengths was obtained using these active bioelectronic chips by simply adjusting the applied field strength. Single base pair mismatch discrimination was achieved in less than 15 seconds with high resolution using an electronic field denaturation approach [107]. In another study, large-scale identification, mapping, and genotyping of SNPs of a 2.3 megabase-long human genomic DNA were performed using a glass-based high-density variation detection chip

[108]. A total of 3241 candidate SNPs were identified and the constructed genetic map shows 2227 locations of these SNPs. Each assay needs more than 15 hours. Again, similar analyses could be performed in a much shorter time if an active chip was used during hybridization [109]. Both oligonucleotide and cDNA has been arrayed on glass substrate to monitor the gene expressions [110, 111]. In one report genome-wide expression monitoring in *Saccharomyces cerevisiae* was conducted using a glass chip with more than 260 000 specifically chosen oligonucleotide probes [110]. Expression levels ranging from less than 0.1 copies to several hundred copies per cell have been measured for cells grown in rich and minimal media. The measurements are quantitative, sensitive, specific, and reproducible. In another study, cDNA microarray containing 9996 elements was made on glass using a robotic arm, and was used for the investigation of the response of fiberblasts to serum. It demonstrated that many features of the transcriptional program could be related to the physiology of the wound repair [111]. Recently, genomic gene expression was analyzed in relation to the response of the Saccharomyces cerevisiae to two antibiotics, amphotericin B, and nystatin. There are approximately 6000 ORFs from Saccharomyces cerevisiae printed on the slide to make the microarray [113]. Also, a microarray with 14 160 human cDNAs was utilized to study the mechanism of Alzheimer's disease. The preliminary results support the amyloid cascade hypothesis as the mechanism of the disease [114].

Analyses of proteins. To bridge genomics and proteomics, protein microarrays are a powerful tool for linking gene expression to molecular binding on a whole-genome level. If differentially expressed genes are discovered through cDNA microarray approach, the same clones can then be examined simultaneously for protein expression in different cellular systems or by *in vitro* transcription/translation. Peptide microarrays were made on active silicon chips [112]. Each microelectrode fabricated on the chip is individually addressable through a CMOS circuitry built also on the silicon substrate. The peptide arrays were made through the electropolymerization process of the pyrrole-modified peptide. The peptide fragments belong to adrenocorticotrophic hormone. Once arrayed, these peptides were examined through immunodetection. In a different study, protein solutions were arrayed onto polyvinylidene difluoride filters at high density by a robotic system [115]. The fabricated protein chips were used for protein expression studies and could also be used for antibody specificity screening against whole libraries of proteins. Techniques developed for deposition of macromolecules onto solid supports include microdispensing [116], electrospray deposition [117], robotic printing [118], stamping [119], inkjet deposition [120], and ion soft-landing [121]. In a recent study, protein microarrays were generated by printing complementary DNAs onto glass slides and then translating target proteins with mammalian reticulocyte lysate. Epitope tags fused to the proteins allowed them to be immobilized in situ. This obviated the need to purify proteins, avoided protein stability problems during storage, and captured sufficient proteins for functional studies. They used the technology to map pairwise interactions among 29 human DNA replication initiation proteins, recapitulate the regulation of Cdt1 binding to select replication proteins, and map its geminin-binding domain [122]. In our laboratory, a protein

chip was developed for doping analysis. The assay was competitive immunoassay-based. Rather high detection sensitivity was obtained with the real athlete urine samples [123].

5.6. System integration

Building lab-on-a-chip systems has now become the central focus in miniaturization of bioanalytical processes. In general, a lab-on-a-chip system should include three representative parts of all biological assays, namely sample processing, biochemical reaction, and detection [124]. Sample handling and manipulation generally includes cell separation, lysis, and DNA/RNA isolation. Biochemical reaction may include various enzymatic reactions such as polymerase chain reaction or proteinase K digestion, and chemical labeling. Detection for reactants has mainly been done through two approaches. One is based on molecular separation using techniques such as capillary electrophoresis or high performance liquid chromatography, and the other is based on affinity binding. The integration of the three steps described above cannot be achieved without the use of microfabricated devices (such as cell separation chip, DNA amplification chip, etc.) and microfluidic control units (such as miniaturized valves and pumps). Several achievements have been made with partial integration of these three key steps including the integration of sample preparation with biochemical reaction [54, 125, 126] and the integration of the biochemical reaction with molecular detection [62, 63, 71, 127, 128]. A complete lab-on-a-chip system has been constructed which clearly demonstrates the possibility of this type of work [64]. Compared to the traditional approaches, a fully integrated portable lab-on-a-chip system has the advantages of reduced contamination, minimal human intervention, mobility, reproducibility, and low consumption of costly reagents and samples. It is anticipated that a completely integrated and self-contained portable lab-on-a-chip will have numerous applications in areas such as point-of-care diagnosis, scene-of-crime identification, outer-space exploitation, on-site agricultural testing, and environmental monitoring. The following summarizes the efforts towards the construction of various lab-on-a-chip systems.

5.6.1. Nucleic acid analysis system

Integration of sample processing and reaction. The separation of white blood cells from red blood cells followed by thermal lysis and PCR amplification was performed in a single silicon-glass chip [54]. The integrated microchip was designed to combine a microfilter and a reactor together. In a different study, isolation of *Escherichia coli* cells from human blood through dielectrophoresis followed by the electronic lysis of the isolated *E. coli* cells was achieved [125]. Moreover, transportation of different cells in microchannels by either electrophoretic pumping or electroosmotic pumping followed by chemical lysis was attempted [126].

Separation-based system. The ultimate goal of developing microchip-based devices is to build a so-called lab-on-a-chip system. Generally, a lab-on-a-chip

system should be composed of three typical steps usually seen with all biological assays, that is, sample preparation, chemical reactions, and detection [124]. Sample preparation generally includes the cell isolation, lysis and DNA/RNA extraction. Chemical reaction usually is represented by various enzymatic reactions such as PCR or proteinase K digestion, or chemical labeling and so forth. Detection for nucleic acids is achieved through two approaches. One is separation-based detection such as capillary electrophoresis or HPLC, and the other is represented by hybridization-based approach. In the following only the efforts devoted for the construction of capillary electrophoresis-based lab-on-a-chip systems are reviewed.

In the integration process of combining sample preparation, chemical reaction and capillary electrophoresis-based detection together, only partial integration has been obtained. The group at University of Pennsylvania led by Wilding and Kricka have reported the first case of performing sample preparation and chemical reaction in one chip [54]. In this report a silicon-glass chip with a microfilter and reactor integrated has been utilized for the isolation of white blood cells from red blood cells followed by a PCR amplification of the DNA released from the isolated white blood cells (Figure 5.4). Using a single bioelectronic chip Cheng et al. have made possible the isolation of *E. coli* cells from human blood through dielectrophoretic separation process followed by the electronic lysis of the isolated cells (Figure 5.5) [125]. Similarly cultured cervical carcinoma cells were isolated from normal human blood cells by this group [59]. Cell transportation in microchannels by electrophoretic pumping and/or electroosmotic pumping has been investigated followed by chemical lysis using SDS buffer [126]. The partial integration of enzymatic reaction and capillary electrophoretic separation has been made on a single glass chip by Jacobson and Ramsey [128]. In this study both the plasmid pBR322 DNA and the restriction enzyme *Hinf* I was electrophoretically pumped into a 0.7 nL reaction chamber where the digestion occurred. The digested DNA fragments were then sized in the capillary etched on the same chip. The entire process was completed in 5 minutes. In a joint study DNA was amplified randomly in a silicon-glass chip using DOP-PCR followed by specific multiplex PCR amplification of dystrophin gene in a second chip. The amplicons were then separated in a glass microchip by capillary electrophoresis (Figure 5.6) [73]. Functional integration of PCR amplification followed by capillary electrophoretic fragment sizing was done by coupling the silicon PCR chip with the glass capillary electrophoresis chip [127]. Using the in-chip polysilicon heater, fast PCR amplification of a β-globin target cloned from M 13 was finished in 15 minutes. The followed capillary electrophoresis separation took approximately 2 minutes. One single glass chip has been fabricated by Ramsey's group to perform combined cell lysis, multiplex PCR, and capillary electrophoresis separation [62, 63]. The PCR, thermal cycling took a couple of hours and the separation of amplicons took about 3 minutes using either HEC or poly(dimethylacrylamide) as sieving gels. Great progress has been made in the fabrication and utilization of a microfabricated silicon chip with integrated metering capability, thermal pump, isothermal reactor, and capillary electrophoresis structure [71]. The amplification of DNA through

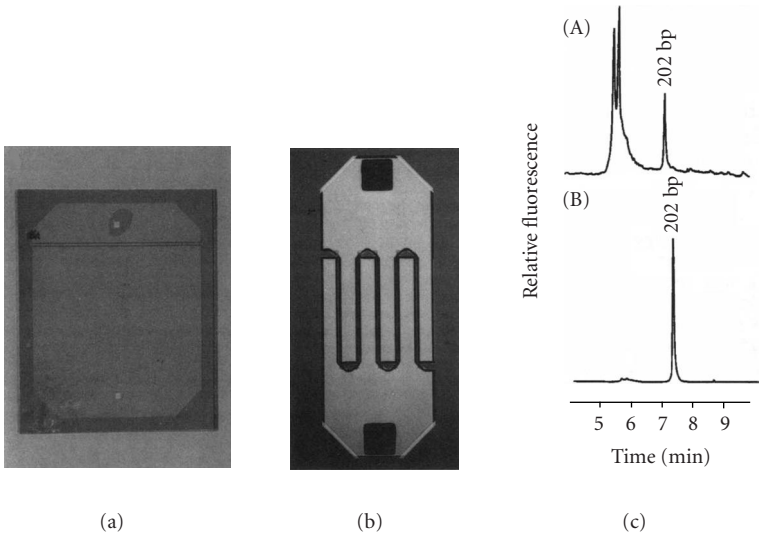


Figure 5.4. Filtration-PCR chip designs and results of direct PCR for dystrophin gene. (a) Integrated filter-PCR chip based on linear weir-type filter in the PCR chamber. (b) Integrated filter-PCR chip based on coiled weir-type filter in the PCR chamber. (c) Electrophoretograms of 202 bp-amplification product (exon 6, dystrophin gene) from direct PCR of DNA in filtered white blood cells using a filter-PCR chip (A) and a positive control (B). A 0.5-μL sample was removed from the chip, diluted with 99.5 μL water, and then injected (5 kV/5 s) onto a 100 μm $\times$ 27 cm-long DB-1-coated capillary in a P/ACE 5050 (run 8 kV/10 min). From [54], with permission.

SDA—an isothermal amplification technique like PCR has been adopted and the amplicon was separated in the microchannel and detected by the integrated photodetector. The total performance of metering the reactants, amplification, and capillary electrophoresis-based separation took approximately 20 minutes (Figure 5.7). This milestone work has shown the possibility in making a complete integrated and self-contained portable device in the near future for many applications such as point-of-care in hospitals, scene-of-crime identification, outerspace exploitation, on-site agricultural testing, and environmental monitoring.

Hybridization-based system. There have been only a few studies about the construction of a lab-on-a-chip through DNA hybridization. Integration of PCR reaction and hybridization was achieved using passive glass chips packaged in a plastic cartridge made of polycarbonate [129]. The fluidic manipulation was achieved using pneumatically actuated valves, pumps, and porous hydrophorbic vents. Once DNA is extracted, the premixed components for PCR were introduced into the inlet port on the cartridge. The amplicons were then allowed to go through a series of reactions inside the cartridge to become fragmented and fluorescently labeled. In the end the labeled DNA targets were detected by hybridization with the oligonucleotide probes attached on a glass microarray chip. The chemical processing stage

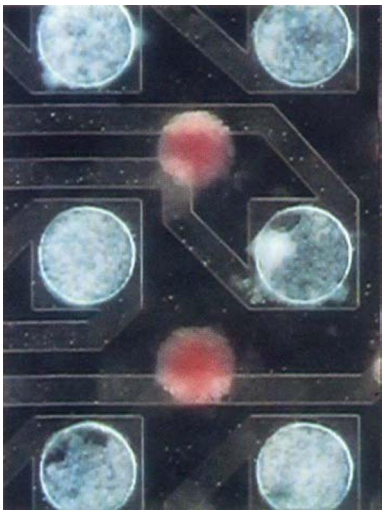


Figure 5.5. Dielectrophoretic separation of *E. coli* from blood cells. The mixture is separated according to charge, bacteria localizing in zones under electric field maxima (white) and erythrocytes localizing in zones under filed minima (red). From [125], with permission.

needs approximately 2–3 hours and the detection via hybridization takes approximately 8 hours. The overall time spent is over 10 hours. The reason for the lengthy processes is as follows. Thermal cycling for PCR accomplished by heating/cooling the plastic cartridge with the thermoelectric devices is not efficient as plastic is not a good thermal material. Furthermore, hybridization on a glass microarray chip takes place by the passive diffusion process and therefore hours of processing time is required. In another system, sample processing and hybridization-based electronic detection has been integrated into a hand-held instrument [130]. This system can process the crude samples such as blood, lyse cells and release DNA. The released DNA molecules are hybridized with the immobilized DNA probes. The DNA probe was attached to the electrode pads on the chip through phenylacetylene polymer. After hybridization, reporter DNA molecules linked to ferrocene redox labels (an amperometric bioelectronic reporter) were added. When voltage is increased an electric current associated with reduction/oxidation of ferrocene labels is detected to differentiate hybridized and unhybridized DNAs at electrodes. The current detection sensitivity is around 10^7 copies. Two issues should be further examined here. First, if the directly released DNA is always used without any amplification stage, complexity reduction in sample components has to be considered for many applications. Second, the hybridization stringency required by point mutation detection and the detection sensitivity for rare events should be ensured.

Progress has been made on the miniaturization of the fluorescence-based optical detection system. A custom-designed integrated circuit containing 4×4 array of phototransistors and on-board signal processing was fabricated [131]. Each

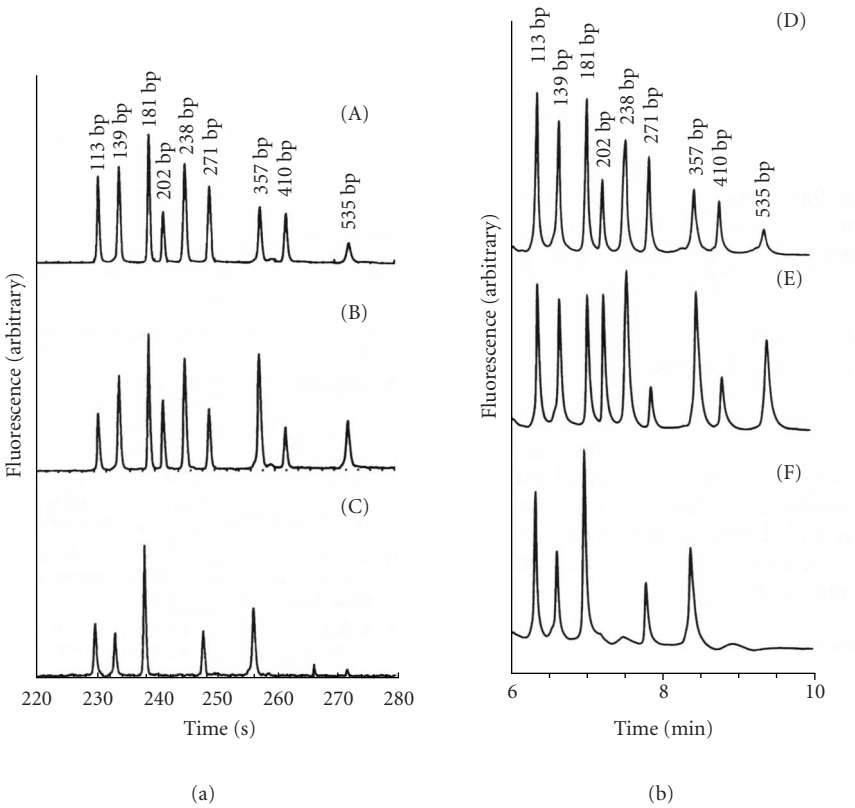


Figure 5.6. Chip CE electropherograms of multiplex PCR amplicons produced using extracted human genomic DNA as template. (A) The amplicons generated in a GeneAmp test tube where a normal human genomic DNA sample was used as template; (B) the amplicons generated in a silicon-glass chip using the same human genomic DNA sample as in (A) was used as template; (C) the amplicons generated in a silicon-glass chip where the human genomic DNA from an affected patient was used as template. Conventional CE electropherograms (D–F) for amplicons used in (A–C), respectively. From [73], with permission.

phototransistor-sensing element fabricated on the integrated circuit chip composed of 220 phototransistor cells connecting in parallel, and can completely convert an optical signal to an electronic signal suitable for data digitization and capture by a computer. When used for the detection of the induced fluorescence signal in DNA microarray, the signals from this amplifier/transistor chip were directly recorded without the need of any electronic interface system or signal amplification device. Such integrated circuit device is useful in constructing a portable lab-on-a-chip system with the capability of detecting multiple DNA targets simultaneously.

A complete sample-to-answer system with portable size and short operation time has been constructed and reported using active bioelectronic chips [64]. To

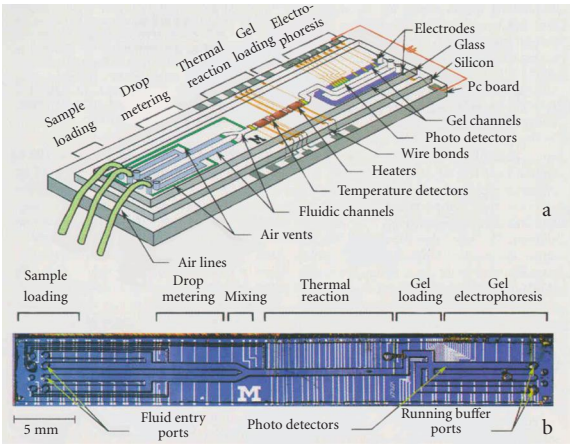
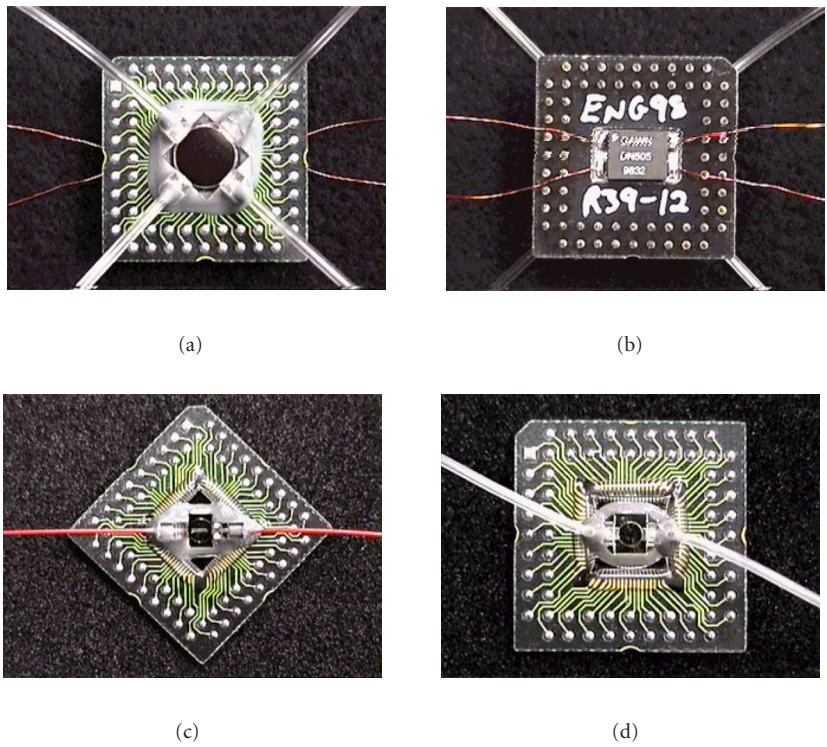


Figure 5.7. Schematic of integrated device with two liquid samples and electrophoresis gel present. The only electronic component not fabricated on the silicon substrate, except for control and data-processing electronics, is an excitation light source placed above the electrophoresis channel. (a) Color code: blue, liquid sample (ready for metering); green, hydrophobic surfaces; purple, polyacrylamide gel. (b) Optical micrograph of the device from above. Wire bonds to the printed circuit board can be seen along the top edge of the device. The blue tint is due to the interference filter reflecting the short-wavelength light. The pressure manifold and buffer wells that fit over the entry holes at each end of the device are not shown. From [71], with permission.

improve the cell isolation efficiency, 100×100 microelectrode array was fabricated on a 1 cm^2 electronic chip to replace the previous 10×10 array. To construct a functional device, a thin layer of sol-gel was first coated on the chip to prevent biomolecules from being damaged and to reduce the nonspecific adsorption of chemical components. Secondly, a flow cell was glued onto the coated chip to facilitate the fluidic manipulation. The machined plastic flow cell has four ports for the introduction of sample and reagents, and also for chemical mixing. This chip assembly has been used for cell separation, cell lysis, and also DNA amplification. To facilitate the SDA reaction a ceramic chip heater is attached to the bottom surface of the electronic chip (Figure 5.8). The details of the fluidic assembly may be found in a report by Cheng et al. [64]. The DNA amplicons obtained in the 100×100 chip was then transported through the connecting tubing to the second chip by a fluidic system with twelve miniature three-way solenoid valves driven by a computer-controlled solenoid pump. The second chip has an array of 5×5 microelectrodes. The agarose coating on the chip works as permeation layer with oligonucleotide probes preimmobilized (through biotin-streptavidin interaction) on the selected microlocations right above the electrodes. When the denatured and desalted amplicons were transported into this chip a DC-based electronic hybridization process [125] was adapted to detect specific marker sequences in the amplified DNA. To enable the detection, a battery-operated 635 nm diode laser and a CCD camera coupled with a set of filters and a zoom lens was used. The

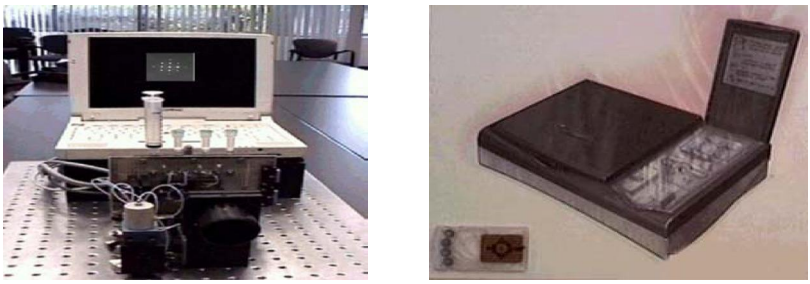


(a) (b)

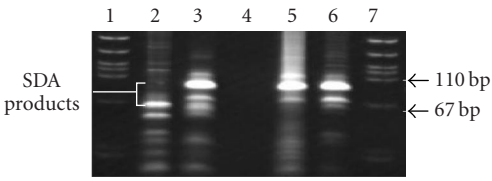
(c) (d)

Figure 5.8. (a) The front view of the packaged bioelectronic chip with 100 × 100 microelectrodes and four-ports flow cell for dielectrophoresis enabled cell separation and SDA-based DNA amplification. (b) The back view of the above-mentioned bioelectronic chip where the miniature ceramic heater was placed tightly against the back side of the silicon chip for providing constant temperatures required by isothermal DNA amplification. (c) The front view of the packaged DNA hybridization chip with 5 × 5 microelectrodes. (d) The close-up of the cartridge bearing the chip for sample preparation and reaction and the chip for hybridization-based DNA analysis (bottom right). These two chips were connected through complex fluidic tubing.

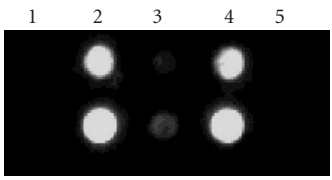
use of a sinusoidal wave signal for both cell separation and electronic hybridization greatly simplified the design of the device. The battery-operated diode laser has a power of 2 mW and an emission wavelength of 635 nm. The fluorescent dye used to label the reporter probe was Bodipy-630. The wavelength of the emission filter is 670 nm. The dichromatic mirror has a wavelength cutoff at 645 nm. With this prototype lab-on-a-chip system, cell separation and lysis process takes 15–25 minutes depending on the complexity of the sample. Other processes including denaturation and desalting, SDA amplification, and hybridization-based detection requires approximately 5, 30, and 5 minutes, respectively. Typically, a complete sample-to-answer process requires a total of approximately 1 hour to complete (Figure 5.9). As the first commercial, analytical instrument based on chip-based capillary electrophoresis technology the Agilent 2100 bioanalyzer has



(a) (b)



(c)



(d)

Figure 5.9. (a) The completed lab-on-a-chip system. (b) Industrial design of the lab-on-a-chip system with the assay chamber lid open. Shown in the bottom left corner is a plastic molded cartridge. (c) The SDA reaction products detected by gel electrophoresis. Note that the SDA reaction yields two specific amplification products, a full-length product, and a sorter, endonuclease-cleaved product. (d) Electronic hybridization of amplification products detected by the CCD-based imaging system used for the prototyping of the portable instrument. The parameters for sine wave applied for each electrode are 1.6 V, 10 HZ, offset +1.2 V for 3 minutes (bottom right). The parameter for DC applied to each electrode is 400 nA for 2 min. From [64], with permission.

proven to be an excellent alternative to messy and labor-intensive gel electrophoresis techniques; delivering fast, automated, high-quality digital data instead of the subjective, time-consuming results associated with gels.

5.6.2. Immunoassay system

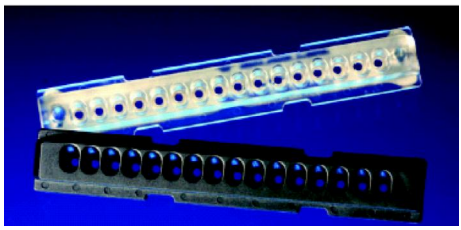
The development of chip-based devices for immunoassay has just started. Using Borofloat glass as substrate a 7.6×7.6 cm device was fabricated to accommodate

the electroosmotic pump, mixer/reactor as well as the electrophoretic microchannels [132]. This functionally integrated device was tested for a competitive assay. Serum sample and theophyl labeled tracer was driven into the first mixer by electroosmotic pumping and becomes mixed. The reactant was then allowed to react with antitheophylline in the second mixer at a fixed ratio. The components from the competitive reaction were finally separated and identified by the capillary electrophoresis on chip. The total time required for reagent mixing with diluted serum samples, immunological reaction, and capillary electrophoresis separation is a few minutes. An MEMS-based generic microfluidic system has been built to facilitate electrochemical immunoassay [32]. The microfluidic system consists of two intersected fluidic paths, one for sampling and one for detection. Magnetic beads, magnetic valves, magnetic pumps, and flow sensors have been adapted or developed for both sampling and manipulating the target biological molecules. This fluidic system was tested for detecting model target molecules called p-Aminophenol using the bead-based sandwich enzyme immunoassay.

5.6.3. Cell analysis system

Optical characterization. The dielectric property of cells has been exploited for cell analysis using a biological microcavity laser system [133]. The vertical cavity surface emitting laser device fabricated in the GaAs is the most important unit of the entire system. When cells were introduced into the chip, different lasing spectra were generated as a result of the dielectric changes among different cells. Submicron-size biochemical sensors and electrodes can be used for the analysis of intracellular parameters (e.g., pH, conductivity) and to detect the presence of cell metabolites (e.g., calcium). The electrochemical signature of peroxynitrite oxidation, an important biologically active species, has been studied using microelectrodes at the single-cell level [134]. A method for preparing platinum electrodes with nanometer dimensions has been reported [135], demonstrating the ability to voltammetrically detect zeptomole quantities of an electroactive species. Recently, an attempt was made to make a micro-ion sensor array to determine intracellular ion concentrations [136].

Electrical characterization of single cells. Chip-based patch clamping has the objective to replace traditional patch electrodes with a planar array of recording interfaces miniaturized on the surface of either a silicon, polymer, or glass substrate. One chip-based device for patch clamping was presented by Schmidt et al. [137] and consists of planar insulating diaphragms on silicon. In this work it was shown that stable gigaohm seals over micrometer-sized holes could be obtained in the time frame of seconds by the electrophoretic self-positioning of charged lipid membranes. Recording chips can be produced in large numbers with defined geometry and material properties by standard silicon technology. Multiple recording sites can be integrated on one single chip because of the small lateral size of the diaphragms. 3D silicon oxide micronozzles integrated into a fluidic device for patch clamping were developed by Lehnert et al. [138]. A cell can be positioned on the nozzle by suction through the hollow nozzle that extends to



(a)

(b)

Figure 5.10. (a) The glass-chip-based patch-clamping devices for high-throughput screening. There are 16 channels on each device capable of analyzing 16 individual cells during each run. The through hole on the glass chip is around 1 μm. (b) The instrument facilitates the fully automated patch-clamping screening processes.

the back of the chip. A microanalysis system for multipurpose electrophysiological analyses has been presented by Han et al. [139]. This system has the capability to perform whole-cell patch clamping, impedance spectroscopy, and general extracellular stimulation/recording using integrated, multielectrode configurations. A high-throughput patch-on-chip device (SealChip 16) was developed by Aviva Bioscience with 16 channels to process 16 single cells in each run (Figure 5.10). The chip-based patch clamping was fully automated with the instrument PatchXpress-7000 provided by Axon Instrument [140].

The loss of physical integrity in the plasma membrane is one of the major indications of cell death. Cell viability is thus usually determined through examination

of membrane integrity with colorometic of fluorescent dyes. Huang et al. [141] developed a new technology that employs a microfabricated device for high-resolution, real-time evaluation of membrane electrical properties of single cells. The chip allows a single cell to be probed with low electrical potentials without introducing membrane damage and permits the corresponding electrical currents flow through that cell to be measured. Electrical resistances of dead (membrane impaired) cells and live cells were found to be significantly different. This suggests that evaluating membrane resistances of individual cells can provide an instant and quantitative measure to determine cell membrane integrity and cell viability of single cells.

The research and development in biochips progressed rapidly. DNA chip technology for mutation detection and gene expression profiling are well established. Now more and more effort is dedicated to the development of chips for the analysis of protein content and cell metabolic substances. For the identification of genomic transcription factor (TF) binding sites *in vivo* in a high-throughput manner, a microarray-based assay of chromatin immunoprecipitation ("ChIP-chip"), also referred to as genome-wide location analysis [19] was developed. ChIP-chip technology is the method that can help to answer which genes are regulated by one specific TF. Following this, enabling multiple TF profiling technologies such as oligonucleotide array-based transcription factor assay (OATFA) may answer which TFs are activated in the cell. Other chips for protein analysis may include protein-spotted microarrays or antibody microarrays for drug screening purposes. But the maintenance of the protein structures on the chip after immobilization remains a challenge. Chips for cell analysis especially single-cell analysis are worthy of watching. Ideally we are hoping to have the electronic measurement and fluorescence assay combined and integrated for cell-based assays. Sensitive detection scheme and nano-sized feature design are essential for the success in the area. System integration for constructing a lab-on-a-chip system is still an ongoing effort, a lot of progress has been made so far but more effort is still needed.

Acknowledgments

This work is funded by the National High-Tech Program of China (no. 2002 AA2Z2011) and Beijing Natural Science Foundation of China (no. H010 210640121).

Bibliography

[1] A. T. Woolley and R. A. Mathies, "Ultra-high-speed DNA fragment separations using microfabricated capillary array electrophoresis chips," *Proc. Natl. Acad. Sci. USA*, vol. 91, no. 24, pp. 11348–11352, 1994.

[2] Z. H. Fan and D. J. Harrison, "Micromachining of capillary electrophoresis injectors and separators on glass chips and evaluation of flow at capillary intersections," *Anal. Chem.*, vol. 66, pp. 177–184, 1994.

[3] P. C. Simpson, A. T. Wooley, and R. A. Mathies, "Microfabrication technology for the production of capillary array electrophoresis chips," *Biomed. Microdevices*, vol. 1, no. 1, pp. 7–26, 1998.

[4] T. Nishimoto, H. Nakanishi, H. Abe, et al., "Microfabricated chips for capillary electrophoresis on quartz glass substrates using a bonding with Hydrofluoric acid," in *Proc. Micro Total Analytical Systems*, D. J. Harrison and A. van den Berg, Eds., Kluwer Academic, Dordrecht, The Netherlands, October 1998.

[5] D. F. Weston, T. Smekal, D. B. Rhine, and J. J. Blackwell, "Fabrication of microfluidic devices in silicon and plastic using plasma etching," *Journal of Vacuum Science and Technology B*, vol. 19, no. 6, pp. 2846–2851, 2001.

[6] X. Li, T. Abe, and M. Esashi, "Deep reactive ion etching of Pyrex glass," in *Proc. IEEE 13th Annual International Conference on Micro Electro Mechanical Systems (MEMS '00)*, pp. 271–276, Miyazaki, Japan, January 2000.

[7] A. Bertsch, S. Heimgartner, P. Cousseau, and P. Renaud, "Static micromixers based on large-scale industrial mixer geometry," *Lab Chip*, vol. 1, no. 1, pp. 56–60, 2001.

[8] K. Ikuta, S. Maruo, T. Fujisawa, and A. Yamada, "Micro concentrator with opto-sense micro reactor for biochemical IC chip family. 3D composite structure and experimental verification," in *Proc. IEEE 12th Annual International Conference on Micro Electro Mechanical Systems (MEMS '99)*, pp. 376–381, Orlando, Fla, USA, January 1999.

[9] H. Wu, T. W. Odom, and G. M. Whitesides, "Reduction photolithography using microlens arrays: applications in gray scale photolithography," *Anal. Chem.*, vol. 74, no. 14, pp. 3267–3273, 2002.

[10] C. J. Hayden, "Three-dimensional excimer laser micromachining using greyscale masks," *J. Micromech. Microeng.*, vol. 13, no. 5, pp. 599–603, 2003.

[11] J. Tien, C. M. Nelson, and C. S. Chen, "Fabrication of aligned microstructures with a single elastomeric stamp," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 4, pp. 1758–1762, 2002.

[12] R. M. McCormick, R. J. Nelson, M. G. Alonso-Amigo, D. J. Benvegnu, and H. H. Hooper, "Microchannel electrophoretic separations of DNA in injection-molded plastic substrates," *Anal. Chem.*, vol. 69, no. 14, pp. 2626–2630, 1997.

[13] J. P. Hulme, P. R. Fielden, and N. J. Goddard, "Fabrication of a spectrophotometric absorbance flow cell using injection-molded plastic," *Anal. Chem.*, vol. 76, no. 1, pp. 238–243, 2004.

[14] C. S. Effenhauser, G. J. M. Bruin, A. Paulus, and M. Ehrat, "Integrated capillary electrophoresis on flexible silicone microdevices: analysis of DNA restriction fragments and detection of single DNA molecules on microchips," *Anal. Chem.*, vol. 69, no. 17, pp. 3451–3457, 1997.

[15] L. Martynova, L. E. Locascio, M. Gaitan, G. W. Kramer, R. G. Christensen, and W. A. MacCrehan, "Fabrication of plastic microfluid channels by imprinting methods," *Anal. Chem.*, vol. 69, no. 23, pp. 4783–4789, 1997.

[16] G.-B. Lee, S.-H. Chen, G.-R. Huang, W.-C. Sung, and Y.-H. Lin, "Microfabricated plastic chips by hot embossing methods and their applications for DNA separation and detection," *Sens. Actuators B Chem.*, vol. 75, pp. 142–148, 2001.

[17] H. Becker, W. Dietz, and P. Dannberg, "Microfluidic manifolds by polymer hot embossing for μ-TAS applications," in *Proc. Micro Total Analytical Systems*, D. J. Harrison and A. van den Berg, Eds., pp. 253–256, Kluwer Academic, Dordrecht, The Netherlands, October 1998.

[18] J. Kameoka, H. G. Craighead, H. Zhang, and J. Henion, "A polymeric microfluidic chip for CE/MS determination of small molecules," *Anal. Chem.*, vol. 73, no. 9, pp. 1935–1941, 2001.

[19] M. A. Roberts, J. S. Rossier, P. Bercier, and H. Girault, "UV laser machined polymer substrates for the development of microdiagnostic systems," *Anal. Chem.*, vol. 69, no. 11, pp. 2035–2042, 1997.

[20] J. A. van Kan, A. A. Bettiol, B. S. Wee, T. C. Sum, S. M. Tang, and F. Watt, "Proton beam micromachining: a new tool for precision three-dimensional microstructures," *Sens. Actuators A Phys.*, vol. 92, no. 1-3, pp. 370–374, 2001.

[21] T. Katoh, N. Nishi, M. Fukagawa, H. Ueno, and S. Sugiyama, "Direct writing for three-dimensional microfabrication using synchrotron radiation etching," *Sens. Actuators A Phys.*, vol. 89, no. 1-2, pp. 10–15, 2001.

[22] S. M. Ford, J. Davies, B. Kar, et al., "Micromachining in plastics using X-ray lithography for the fabrication of microelectrophoresis devices," *J. Biomech. Eng.*, vol. 121, no. 1, pp. 13–21, 1999.

[23] M. Schena, *Microarray Analysis*, John Wiley & Sons, New York, NY, USA, 2003.

[24] K. L. Michael, L. C. Taylor, S. L. Schultz, and D. R. Walt, "Randomly ordered addressable high-density optical sensor arrays," *Anal. Chem.*, vol. 70, no. 7, pp. 1242–1248, 1998.

[25] J. M. Yeakley, J. B. Fan, D. Doucet, et al., "Profiling alternative splicing on fiber-optic arrays," *Nat. Biotechnol.*, vol. 20, no. 4, pp. 353–358, 2002.

[26] S. Singh-Gasson, R. D. Green, Y. Yue, et al., "Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array," *Nat. Biotechnol.*, vol. 17, no. 10, pp. 974–978, 1999.

[27] J. Sebat, B. Lakshmi, J. Troge, et al., "Large-scale copy number polymorphism in the human genome," *Science*, vol. 305, no. 5683, pp. 525–528, 2004.

[28] A. P. Blanchard, R. J. Kaiser, and L. E. Hood, "High-density oligonucleotide arrays," *Biosens. Bioelectron.*, vol. 11, pp. 687–690, 1996.

[29] A. Unami, Y. Shinohara, K. Kajimoto, and Y. Baba, "Comparison of gene expression profiles between white and brown adipose tissues of rat by microarray analysis," *Biochem. Pharmacol.*, vol. 67, no. 3, pp. 555–564, 2004.

[30] C. D. Bevan and I. M. Mutton, "Use of freeze-thaw flow management for controlling and switching fluid flow in capillary tubes," *Anal. Chem.*, vol. 67, no. 8, pp. 1470–1473, 1995.

[31] B. Löchel, A. Maciossek, H. J. Quenzer, B. Wagner, and G. Engelmann, "Magnetically driven microstructures fabricated with multilayer electroplating," *Sens. Actuators A Phys.*, vol. 46-47, pp. 98–103, 1995.

[32] C. H. Ahn, T. Henderson, W. Heineman, and B. Halsall, "Development of a generic microfluidic system for electrochemical immunoassay-based remote bio/chemical sensors," in *Proc. Micro Total Analytical Systems*, D. J. Harrison and A. van den Berg, Eds., pp. 225–230, Kluwer Academic, Dordrecht, The Netherlands, October 1998.

[33] H. Hartshorne, Y. B. Ning, W. E. Lee, and C. Backhouse, "Development of microfabricated valves for μTAS," in *Proc. Micro Total Analytical Systems*, D. J. Harrison and A. van den Berg, Eds., pp. 379–381, Kluwer Academic, Dordrecht, The Netherlands, October 1998.

[34] G. Blankenstein and U. D. Larsen, "Modular concept of a laboratory on a chip for chemical and biochemical analysis," *Biosens. Bioelectron.*, vol. 13, pp. 427–438, 1998.

[35] R. B. M. Schasfoort, S. Schlautmann, J. Hendrikse, and A. van den Berg, "Field-effect flow control for microfabricated fluidic networks," *Science*, vol. 286, no. 5441, pp. 942–945, 1999.

[36] R. F. Ismagilov, D. Rosmarin, P. J. A. Kenis, et al., "Pressure-driven laminar flow in tangential microchannels: an elastomeric microfluidic switch," *Anal. Chem.*, vol. 73, no. 19, pp. 4682–4687, 2001.

[37] M. E. Harmon, M. Tang, and C. W. Frank, "A microfluidic actuator based on thermoresponsive hydrogels," *Polymer*, vol. 44, no. 16, pp. 4547–4556, 2003.

[38] W. H. Grover, A. M. Skelley, C. N. Liu, E. T. Lagally, and R. A. Mathies, "Monolithic membrane valves and diaphragm pumps for practical large-scale integration into glass microfluidic devices," *Sens. Actuators B Chem.*, vol. 89, no. 3, pp. 315–323, 2003.

[39] S. E. McBride, R. M. Moroney, and W. Chiang, "Electrohydrodynamic pumps for high-density microfluidic arrays," in *Proc. Micro Total Analytical Systems*, D. J. Harrison and A. van den Berg, Eds., pp. 45–48, Kluwer Academic, Dordrecht, The Netherlands, October 1998.

[40] A. V. Lemoff and A. P. Lee, "An AC magnetohydrodynamic micropump," *Sens. Actuators B Chem.*, vol. 63, pp. 178–185, 2000.

[41] K. Seller, Z. H. Fan, K. Fluri, and D. J. Harrison, "Electroosmotic pumping and valveless control of fluid flow within a manifold of capillaries on a glass chip," *Anal. Chem.*, vol. 66, pp. 3485–3491, 1994.

[42] T. Müller, W. M. Arnold, T. Schnelle, R. Hagedorn, G. Fuhr, and U. Zimmermann, "A traveling-wave micropump for aqueous solutions: comparison of 1 g and microgram results," *Electrophoresis*, vol. 14, no. 8, pp. 764–772, 1993.

[43] M. A. Burns, C. H. Mastrangelo, T. S. Sammarco, et al., "Microfabricated structures for integrated DNA analysis," *Proc. Natl. Acad. Sci. USA*, vol. 93, no. 11, pp. 5556–5561, 1996.

[44] S. Shoji and M. Esashi, "Microfabrication and microsensors," *Appl. Biochem. Biotechnol.*, vol. 41, no. 1-2, pp. 21–34, 1993.

[45] S. Shoji, M. Esashi, B. van der Schoot, and N. de Rooij, "A study of a high-pressure micropump for integrated chemical analysing system," *Sens. Actuators A Phys.*, vol. 32, pp. 335–339, 1992.

[46] A. Olsson, G. Stemme, and E. Stemme, "A valve-less planar fluid pump with two pump chambers," *Sens. Actuators A Phys.*, vol. 46-47, pp. 549–556, 1995.

[47] P. Woias, R. Linnemann, M. Richter, A. Leistner, and B. Hillerich, "A silicon micropump with a high bubble tolerance and self-priming capability," in *Proc. Micro Total Analytical Systems*, D. J. Harrison and A. van den Berg, Eds., pp. 383–386, Kluwer Academic, Dordrecht, The Netherlands, October 1998.

[48] S. Bohm, M. Dierselhuis, W. Olthuis, and P. Bergveld, "Manufacturing of self-priming plastic micropumps," in *Proc. Micro Total Analytical Systems*, D. J. Harrison and A. van den Berg, Eds., pp. 391–394, Kluwer Academic, Dordrecht, The Netherlands, October 1998.

[49] C. Liu, M. Guo, X. Chen, and J. Cheng, "Low voltage driven miniaturized pump with high back pressure," in *MicroFluidics, BioMEMS, and Medical Microsystems (MF '03)*, H. Becker and P. Woias, Eds., vol. 4982 of *Proc. SPIE*, pp. 344–355, San Jose, Calif, USA, 2003.

[50] Y. J. Song and T. S. Zhao, "Modelling and test of a thermally-driven phase-change nonmechanical micropump," *J. Micromech. Microeng.*, vol. 11, no. 6, pp. 713–719, 2001.

[51] B. S. Gallardo, V. K. Gupta, F. D. Eagerton, et al., "Electrochemical principles for active control of liquids on submillimeter scales," *Science*, vol. 283, no. 5398, pp. 57–60, 1999.

[52] M. W. J. Prins, W. J. J. Welters, and J. W. Weekamp, "Fluid control in multichannel structures by electrocapillary pressure," *Science*, vol. 291, no. 5502, pp. 277–280, 2001.

[53] R. H. Carlson, C. Gabel, S. Chan, and R. H. Austin, "Activation and sorting of human white blood cells," *Biomed. Microdevices*, vol. 1, no. 1, pp. 39–47, 1998.

[54] P. Wilding, L. J. Kricka, J. Cheng, G. E. Hvichia, M. A. Shoffner, and P. Fortina, "Integrated cell isolation and polymerase chain reaction analysis using silicon microfilter chambers," *Anal. Biochem.*, vol. 257, no. 2, pp. 95–100, 1998.

[55] G. Blankenstein, "Microfabricated flow system for magnetic cell and particle separation," in *Scientific and Clinical Applications of Magnetic Carriers*, U. Häfeli, W. Schütt, J. Teller, et al., Eds., pp. 223–246, Plenum Press, New York, USA, 1996.

[56] X. B. Wang, Y. Huang, X. Wang, F. F. Becker, and P. R. Gascoyne, "Dielectrophoretic manipulation of cells with spiral electrodes," *Biophys. J.*, vol. 72, no. 4, pp. 1887–1899, 1997.

[57] S. Fiedler, S. G. Shirley, T. Schnelle, and G. Fuhr, "Dielectrophoretic sorting of particles and cells in a microsystem," *Anal. Chem.*, vol. 70, no. 9, pp. 1909–1915, 1998.

[58] T. Schnelle, T. Muller, S. Fiedler, et al., "Trapping of viruses in high-frequency electric field cages," *Naturwissenschaften*, vol. 83, no. 4, pp. 172–176, 1996.

[59] J. Cheng, E. L. Sheldon, L. Wu, M. J. Heller, and J. P. O'Connell, "Isolation of cultured cervical carcinoma cells mixed with peripheral blood cells on a bioelectronic chip," *Anal. Chem.*, vol. 70, no. 11, pp. 2321–2326, 1998.

[60] L. Wang, M. Guo, C. Huang, and J. Cheng, "Continuous cell separation by chip-based traveling-wave dielectrophoresis and laminar flow," in *Proc. Micro Total Analytical Systems (2003)*, M. A. Northrup, K. F. Jensen, and D. J. Harrison, Eds., pp. 299–302, Transducers Research Foundation, Cleveland Heights, Ohio, USA, October 2003.

[61] M. U. Kopp, A. J. Mello, and A. Manz, "Chemical amplification: continuous-flow PCR on a chip," *Science*, vol. 280, no. 5366, pp. 1046–1048, 1998.

[62] L. C. Waters, S. C. Jacobson, N. Kroutchinina, J. Khandurina, R. S. Foote, and J. M. Ramsey, "Microchip device for cell lysis, multiplex PCR amplification, and electrophoretic sizing," *Anal. Chem.*, vol. 70, no. 1, pp. 158–162, 1998.

[63] L. C. Waters, S. C. Jacobson, N. Kroutchinina, J. Khandurina, R. S. Foote, and J. M. Ramsey, "Multiple sample PCR amplification and electrophoretic analysis on a microchip," *Anal. Chem.*, vol. 70, no. 24, pp. 5172–5176, 1998.

[64] J. Cheng, L. Wu, J. Diver, et al., "Fluorescent imaging of cells and nucleic acids in bioelectronic chips," in *Biomedical Imaging: Reporters, Dyes, and Instrumentation*, D. J. Bornhop, C. H. Contag, and E. M. Sevick-Muraca, Eds., vol. 3600 of *Proc. SPIE*, pp. 23–28, San Jose, Calif, USA, 1999.

[65] P. Belgrader, J. K. Smith, V. W. Weedn, and M. A. Northrup, "Rapid PCR for identity testing using a battery-powered miniature thermal cycler," *J. Forensic. Sci.*, vol. 43, no. 2, pp. 315–319, 1998.

[66] R. P. Oda, M. A. Strausbauch, A. F. R. Huhmer, et al., "Infrared-mediated thermocycling for ultrafast polymerase chain reaction amplification of DNA," *Anal. Chem.*, vol. 70, no. 20, pp. 4361–4368, 1998.

[67] T. B. Taylor, E. S. Winn-Deen, E. Picozza, T. M. Woudenberg, and M. Albin, "Optimization of the performance of the polymerase chain reaction in silicon-based microstructures," *Nucleic Acids Res.*, vol. 25, no. 15, pp. 3164–3168, 1997.

[68] T. B. Taylor, P. M. John, and M. Albin, "Micro-genetic analysis systems," in *Proc. Micro Total Analytical Systems*, D. J. Harrison and A. van den Berg, Eds., pp. 261–266, Kluwer Academic, Dordrecht, The Netherlands, October 1998.

[69] M. A. Shoffner, J. Cheng, G. E. Hvichia, L. J. Kricka, and P. Wilding, "Chip PCR. I. Surface passivation of microfabricated silicon-glass chips for PCR," *Nucleic Acids Res.*, vol. 24, no. 2, pp. 375–379, 1996.

[70] J. Cheng, M. A. Shoffner, K. R. Mitchelson, L. J. Kricka, and P. Wilding, "Analysis of ligase chain reaction products amplified in a silicon-glass chip using capillary electrophoresis," *J. Chromatogr. A.*, vol. 732, no. 1, pp. 151–158, 1996.

[71] M. A. Burns, B. N. Johnson, S. N. Brahmasandra, et al., "An integrated nanoliter DNA analysis device," *Science*, vol. 282, no. 5388, pp. 484–487, 1998.

[72] J. Cheng, M. A. Shoffner, G. E. Hvichia, L. J. Kricka, and P. Wilding, "Chip PCR. II. Investigation of different PCR amplification systems in microfabricated silicon-glass chips," *Nucleic Acids Res.*, vol. 24, no. 2, pp. 380–385, 1996.

[73] J. Cheng, L. C. Waters, P. Fortina, et al., "Degenerate oligonucleotide primed-polymerase chain reaction and capillary electrophoretic analysis of human DNA on microchip-based devices," *Anal. Biochem.*, vol. 257, no. 2, pp. 101–106, 1998.

[74] J. Cheng, M. A. Shoffner, Q. Zhang, L. J. Kricka, and P. Wilding, "Examination of the microchip RT-PCR products using entangled solution capillary electrophoresis (ESCE) with laser induced fluorescence detection (LIF)," in *Proc. 6th Annual Frederick Conference on Capillary Electrophoresis*, Frederick, Md, USA, October 1995.

[75] K. Shen, X. Chen, M. Guo, and J. Cheng, "A microchip-based PCR device using flexible printed circuit technology," to appear in Sens. Actuators B Chem.

[76] K. Fluri, G. Fitzpatrick, N. Chiem, and D. J. Harrison, "Integrated capillary electrophoresis devices with an efficient postcolumn reactor in planar quartz and glass chips," *Anal. Chem.*, vol. 68, pp. 4285–4290, 1996.

[77] A. G. Hadd, D. E. Raymond, J. W. Halliwell, S. C. Jacobson, and J. M. Ramsey, "Microchip device for performing enzyme assays," *Anal. Chem.*, vol. 69, no. 17, pp. 3407–3412, 1997.

[78] D. J. Harrison, K. Fluri, K. Seiler, Z. Fan, C. S. Effenhauser, and A. Manz, "Micromachining a miniaturized capillary electrophoresis-based chemical analysis system on a chip," *Science*, vol. 261, pp. 895–896, 1993.

[79] J. C. Fister, S. C. Jacobson, L. M. Davis, and J. M. Ramsey, "Counting single chromophore molecules for ultrasensitive analysis and separations on microchip devices," *Anal. Chem.*, vol. 70, no. 3, pp. 431–437, 1998.

[80] S. C. Jacobson, C. T. Culbertson, J. E. Daler, and J. M. Ramsey, "Microchip structures for submillisecond electrophoresis," *Anal. Chem.*, vol. 70, no. 16, pp. 3476–3480, 1998.

[81] C. S. Effenhauser, A. Paulus, A. Manz, and H. M. Widmer, "High speed separation of antisense oligonucleotides on a micromachined capillary electrophoresis device," *Anal. Chem.*, vol. 66, no. 18, pp. 2949–2953, 1994.

[82] S. Terabe, K. Otsuka, K. Ichikawa, A. Tsuchiya, and T. Ando, "Electrokinetic separations with micellar solutions and open-tubular capillary," *Anal. Chem.*, vol. 56, no. 1, pp. 111–113, 1984.

[83] A. W. Moore Jr., S. C. Jacobson, and J. M. Ramsey, "Microchip separations of neutral species via micellar electrokinetic capillary chromatography," *Anal. Chem.*, vol. 67, pp. 4184–4189, 1995.

[84] I. Rodriguez, H. K. Lee, and S. F. Li, "Microchannel electrophoretic separation of biogenic amines by micellar electrokinetic chromatography," *Electrophoresis*, vol. 20, no. 1, pp. 118–126, 1999.

[85] F. von Heeren, E. Verpoorte, A. Manz, and W. Thormann, "Micellar electrokinetic chromatography separations and analyses of biological samples on a cyclic planar microstructure," *Anal. Chem.*, vol. 68, no. 13, pp. 2044–2053, 1996.

[86] P. A. Walker III, M. D. Morris, M. A. Burns, and B. N. Johnson, "Isotachophoretic separations on a microchip. Normal Raman spectroscopy detection," *Anal. Chem.*, vol. 70, no. 18, pp. 3766–3769, 1998.

[87] O. Hofmann, D. Che, K. A. Cruickshank, and U. R. Muller, "Adaptation of capillary isoelectric focusing to microchannels on a glass chip," *Anal. Chem.*, vol. 71, no. 3, pp. 678–686, 1999.

[88] S. C. Jacobson, R. Hergenroder, L. B. Koutny, and J. M. Ramsey, "Open-channel electrochromatography on a microchip," *Anal. Chem.*, vol. 66, pp. 2369–2373, 1994.

[89] J. P. Kutter, S. C. Jacobson, N. Matsubara, and J. M. Ramsey, "Solvent-programmed microchip open-channel electrochromatography," *Anal. Chem.*, vol. 70, no. 15, pp. 3291–3297, 1998.

[90] D. E. Raymond, A. Manz, and H. M. Widmer, "Continuous sample pretreatment using a free-flow electrophoresis device integrated onto a silicon chip," *Anal. Chem.*, vol. 66, pp. 2858–2865, 1994.

[91] D. E. Raymond, A. Manz, and H. M. Widmer, "Continuous separation of high molecular weight compounds using a microliter volume free-flow electrophoresis microstructure," *Anal. Chem.*, vol. 68, no. 15, pp. 2515–2522, 1996.

[92] Y. Huang, X. B. Wang, F. F. Becker, and P. R. C. Gascoyne, "Introducing dielectrophoresis as a new force field for field-flow fractionation," *Biophys. J.*, vol. 73, no. 2, pp. 1118–1129, 1997.

[93] D. Ertas, "Lateral separation of macromolecules and polyelectrolytes in microlithographic arrays," *Phys. Rev. Lett.*, vol. 80, no. 7, pp. 1548–1551, 1998.

[94] T. A. J. Duke and R. H. Austin, "Microfabricated sieve for the continuous sorting of macromolecules," *Phys. Rev. Lett.*, vol. 80, no. 7, pp. 1552–1555, 1998.

[95] T. A. J. Duke, R. H. Austin, E. C. Cox, and S. S. Chan, "Pulsed-field electrophoresis in microlithographic arrays," *Electrophoresis*, vol. 17, no. 6, pp. 1075–1079, 1996.

[96] A. T. Woolley and R. A. Mathies, "Ultra-high-speed DNA fragment separations using microfabricated capillary array electrophoresis chips," *Proc. Natl. Acad. Sci. USA*, vol. 91, no. 24, pp. 11348–11352, 1994.

[97] D. Schmalzing, L. Koutny, A. Adourian, P. Belgrader, P. Matsudaira, and D. Ehrlich, "DNA typing in thirty seconds with a microfabricated device," *Proc. Natl. Acad. Sci. USA*, vol. 94, no. 19, pp. 10273–10278, 1997.

[98] A. T. Woolley, G. F. Sensabaugh, and R. A. Mathies, "High-speed DNA genotyping using microfabricated capillary array electrophoresis chips," *Anal. Chem.*, vol. 69, no. 11, pp. 2181–2186, 1997.

[99] M. Ogura, Y. Agata, K. Watanabe, et al., "RNA chip: quality assessment of RNA by microchannel linear gel electrophoresis in injection-molded plastic chips," *Clin. Chem.*, vol. 44, no. 11, pp. 2249–2255, 1998.

[100] P. Liu, W. Xing, D. Liang, G. Huang, Y. Zhou, and J. Cheng, "Fast screening of single-nucleotide polymorphisms using chip-based temperature gradient capillary electrophoresis," *Anal. Lett.*, vol. 36, no. 13, pp. 2819–2830, 2003.

[101] A. T. Woolley and R. A. Mathies, "Ultra-high-speed DNA sequencing by using capillary electrophoresis chips," *Anal. Chem.*, vol. 67, no. 20, pp. 3676–3680, 1995.

[102] S. Liu, Y. Shi, W. W. Ja, and R. A. Mathies, "Optimization of high-speed DNA sequencing on microfabricated capillary electrophoresis channels," *Anal. Chem.*, vol. 71, no. 3, pp. 566–573, 1999.

[103] J. Kling, "Ultrafast DNA sequencing," *Nat. Biotechnol.*, vol. 21, no. 12, pp. 1425–1427, 2003.

[104] L. B. Koutny, D. Schmalzing, T. A. Taylor, and M. Fuchs, "Microchip electrophoretic immunoassay for serum cortisol," *Anal. Chem.*, vol. 68, no. 1, pp. 18–22, 1996.

[105] N. Chiem and D. J. Harrison, "Microchip-based capillary electrophoresis for immunoassays: analysis of monoclonal antibodies and theophylline," *Anal. Chem.*, vol. 69, no. 3, pp. 373–378, 1997.

[106] J. G. Hacia, L. C. Brody, M. S. Chee, S. P. Fodor, and F. S. Collins, "Detection of heterozygous mutations in BRCA1 using high density oligonucleotide arrays and two-colour fluorescence analysis," *Nat. Genet.*, vol. 14, no. 4, pp. 441–447, 1996.

[107] R. G. Sosnowski, E. Tu, W. F. Butler, J. P. O'Connell, and M. J. Heller, "Rapid determination of single base mismatch mutations in DNA hybrids by direct electric field control," *Proc. Natl. Acad. Sci. USA*, vol. 94, no. 4, pp. 1119–1123, 1997.

[108] D. G. Wang, J. B. Fan, C. J. Siao, et al., "Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome," *Science*, vol. 280, no. 5366, pp. 1077–1082, 1998.

[109] P. N. Gilles, D. J. Wu, C. B. Foster, P. J. Dillon, and S. J. Chanock, "Single nucleotide polymorphic discrimination by an electronic dot blot assay on semiconductor microchips," *Nat. Biotechnol.*, vol. 17, no. 4, pp. 365–370, 1999.

[110] L. Wodicka, H. Dong, M. Mittmann, M. H. Ho, and D. J. Lockhart, "Genome-wide expression monitoring in *Saccharomyces cerevisiae*," *Nat. Biotechnol.*, vol. 15, no. 13, pp. 1359–1367, 1997.

[111] V. R. Lyer, M. B. Eisen, D. T. Ross, et al., "The transcriptional program in the response of human fibroblasts to serum," *Science*, vol. 283, no. 5398, pp. 83–87, 1999.

[112] T. Livache, H. Bazin, P. Caillat, and A. Roget, "Electroconducting polymers for the construction of DNA or peptide arrays on silicon chips," *Biosens. Bioelectron.*, vol. 13, no. 6, pp. 629–634, 1998.

[113] L. Zhang, Y. Zhang, Y. Zhou, S. An, Y. Zhou, and J. Cheng, "Response of gene expression in Saccharomyces cerevisiae to amphotericin B and nystatin measured by microarrays," *J. Antimicrob. Chemother.*, vol. 49, no. 6, pp. 905–915, 2002.

[114] G. Wang, Y. Zhang, B. Chen, and J. Cheng, "Preliminary studies on Alzheimer's disease using cDNA microarrays," *Mech. Ageing Dev.*, vol. 124, no. 1, pp. 115–124, 2003.

[115] A. Lueking, M. Horn, H. Eickhoff, K. Büssow, H. Lehrach, and G. Walter, "Protein microarrays for gene expression and antibody screening," *Anal. Biochem.*, vol. 270, no. 1, pp. 103–111, 1999.

[116] T. Laurell, L. Wallman, and J. Nilsson, "Design and development of a silicon microfabricated flow-through dispenser for on-line picolitre sample handling," *J. Micromech. Microeng.*, vol. 9, no. 4, pp. 369–376, 1999.

[117] V. N. Morozov and T. Y. Morozova, "Electrospray deposition as a method for mass fabrication of mono- and multicomponent microarrays of biological and biologically active substances," *Anal. Chem.*, vol. 71, no. 15, pp. 3110–3117, 1999.

[118] G. MacBeath and S. L. Schreiber, "Printing proteins as microarrays for high-throughput function determination," *Science*, vol. 289, no. 5485, pp. 1760–1763, 2000.

[119] M. A. Markowitz, D. C. Turner, B. D. Martin, and B. P. Gaber, "Diffusion and transfer of antibody proteins from a sugar-based hydrogel," *Appl. Biochem. Biotechnol.*, vol. 68, no. 1-2, pp. 57–68, 1997.

[120] A. Roda, M. Guardigli, C. Russo, P. Pasini, and M. Baraldini, "Protein microdeposition using a conventional ink-jet printer," *Biotechniques*, vol. 28, no. 3, pp. 492–496, 2000.

[121] R. G. Cooks and Z. Ouyang, "System and method for the preparation of arrays of biological or other molecules", US Patent, Pub. App. no. 20030226963, 2003.

[122] N. Ramachandran, E. Hainsworth, B. Bhullar, et al., "Self-assembling protein microarrays," *Science*, vol. 305, no. 5680, pp. 86–90, 2004.

[123] H. Du, M. Wu, W. Yang, et al., "Development of miniaturized competitive immunoassays on a protein chip as a screening tool for drugs," *Clin. Chem.*, vol. 51, no. 2, pp. 368–375, 2005.

[124] J. Cheng, P. Fortina, S. Surrey, L. J. Kricka, and P. Wilding, "Microchip-based devices for molecular diagnosis of genetic diseases," *Mol. Diagn.*, vol. 1, no. 3, pp. 183–200, 1996.

[125] J. Cheng, E. L. Sheldon, L. Wu, et al., "Preparation and hybridization analysis of DNA/RNA from *E. coli* on microfabricated bioelectronic chips," *Nat. Biotechnol.*, vol. 16, no. 6, pp. 541–546, 1998.

[126] P. C. Li and D. J. Harrison, "Transport, manipulation, and reaction of biological cells on-chip using electrokinetic effects," *Anal. Chem.*, vol. 69, no. 8, pp. 1564–1568, 1997.
[127] A. T. Woolley, D. Hadley, P. Landre, A. J. deMello, R. A. Mathies, and M. A. Northrup, "Functional integration of PCR amplification and capillary electrophoresis in a microfabricated DNA analysis device," *Anal. Chem.*, vol. 68, no. 23, pp. 4081–4086, 1996.
[128] S. C. Jacobson and J. M. Ramsey, "Integrated microdevice for DNA restriction fragment analysis," *Anal. Chem.*, vol. 68, no. 5, pp. 720–723, 1996.
[129] R. C. Anderson, G. J. Bogdan, Z. Barniv, T. D. Dawes, J. Winkler, and K. Roy, "Microfluidic biochemical analysis system," in *Proc. IEEE International Conference on Solid State Sensors and Actuators, Transducers '97*, vol. 1, pp. 477–480, Chicago, Ill, USA, June 1997.
[130] J. Hodgson, "Shrinking DNA diagnostics to fill the markets of the future," *Nat. Biotechnol.*, vol. 16, no. 8, pp. 725–727, 1998.
[131] T. Vo-Dinh, J. P. Alarie, N. Isola, D. Landis, A. L. Wintenberg, and M. N. Ericson, "DNA biochip using a phototransistor integrated circuit," *Anal. Chem.*, vol. 71, no. 2, pp. 358–363, 1999.
[132] N. H. Chiem and D. J. Harrison, "Microchip systems for immunoassay: an integrated immunoreactor with electrophoretic separation for serum theophylline determination," *Clin. Chem.*, vol. 44, no. 3, pp. 591–598, 1998.
[133] P. L. Gourley, "Semiconductor microlasers: a new approach to cell-structure analysis," *Nat. Med.*, vol. 2, no. 8, pp. 942–944, 1996.
[134] C. Amatore, S. Arbault, D. Bruce, P. de Oliveira, L. M. Erard, and M. Vuillaume, "Characterization of the electrochemical oxidation of peroxynitrite: relevance to oxidative stress bursts measured at the single cell level," *Chemistry*, vol. 7, no. 19, pp. 4171–4179, 2001.
[135] J. J. Watkins, J. Chen, H. S. White, H. D. Abruna, E. Maisonhaute, and C. Amatore, "Zeptomole voltammetric detection and electron-transfer rate measurements using platinum electrodes of nanometer dimensions," *Anal. Chem.*, vol. 75, no. 16, pp. 3962–3971, 2003.
[136] O. T. Guenat, X. J. Wang, J.-F. Dufour, et al., "Ion-selective microelectrode array for intracellular detection on chip," in *Proc. IEEE 12th International Conference on Solid-State Sensors, Actuators and Microsystems, Transducers '03*, vol. 2, pp. 1063–1066, Switzerland, June 2003.
[137] C. Schmidt, M. Mayer, and H. Vogel, "A chip-based biosensor for the functional analysis of single ion channels," *Angew. Chem. Int. Ed. Engl.*, vol. 39, no. 17, pp. 3137–3140, 2000.
[138] T. Lehnert, M. A. M. Gijs, R. Netzer, and U. Bischoff, "Realization of hollow SiO_2 micronozzles for electrical measurements on living cells," *Appl. Phys. Lett.*, vol. 81, no. 26, pp. 5063–5065, 2002.
[139] A. Han, E. Moss, R. D. Rabbitt, and A. B. Frazier, "A multi-purpose micro system for electrophysiological analyses of single cells," in *Proc. Micro Total Analysis Systems 2002*, pp. 805–807, Nara, Japan, November 2002.
[140] C. Smith, "Drug target identification: a question of biology," *Nature*, vol. 428, no. 6979, pp. 225–231, 2004.
[141] Y. Huang, N. Sekhon, J. Borninski, N. Chen, and B. Rubinsky, "Instantaneous, quantitative single-cell viability assessment by electrical evaluation of cell membrane integrity with microfabricated devices," *Sens. Actuators A Phys.*, vol. 105, no. 1, pp. 31–39, 2003.

Lei Wang: Department of Biological Sciences and Biotechnology, Tsinghua University, Beijing 100084, China

Email: wang-101@mails.tsinghua.edu.cn

Hongying Yin: Department of Biological Sciences and Biotechnology, Tsinghua University, Beijing 100084, China

Email: yinhy01@mails.tsinghua.edu.cn

Jing Cheng: Department of Biological Sciences and Biotechnology, Tsinghua University, Beijing 100084, China; National Engineering Research Center for Bejing Biochip Technology, Beijing 102206, China

Email: jcheng@tsinghua.edu.cn

6 Modeling and simulation of genetic regulatory networks by ordinary differential equations

Hidde de Jong and Johannes Geiselmann

A remarkable development in molecular biology has been the recent upscaling to the genomic level of its experimental methods. These methods produce, on a routine basis, enormous amounts of data on different aspects of the cell. A large part of the experimental data available today concern genetic regulatory networks underlying the functioning and differentiation of cells. In addition to high-throughput experimental methods, mathematical and computational approaches are indispensable for analyzing these networks of genes, proteins, small molecules, and their mutual interactions. In this chapter, we review methods for the modeling and simulation of genetic regulatory networks. A large number of approaches have been proposed in the literature, based on such formalisms as graphs, Boolean networks, differential equations, and stochastic master equations. We restrict the discussion here to ordinary differential equation models, which is probably the most widely used formalism. In particular, we compare nonlinear, linear, and piecewise linear differential equations, illustrating the application of these models by means of concrete examples taken from the literature.

6.1. Introduction

A remarkable development in molecular biology today is the upscaling to the genomic level of its experimental methods. Hardly imaginable only 20 years ago, the sequencing of complete genomes has become a routine job, highly automated and executed in a quasi-industrial environment. The miniaturization of techniques for the hybridization of labeled nucleic acids in solution to DNA molecules attached to a surface has given rise to DNA microarrays, tools for measuring the level of gene expression in a massively parallel way [1]. The development of proteomic methods based on two-dimensional gel electrophoresis, mass spectrometry, and the double-hybrid system allows the identification of proteins and their interactions on a genomic scale [2].

These novel methods in genomics produce enormous amounts of data about different aspects of the cell. On one hand, they allow the identification of interactions between the genes of an organism, its proteins, metabolites, and other small

molecules, thus mapping the structure of its interaction networks. On the other hand, they are able to detect the evolution of the state of the cell, that is, the temporal variation of the concentration and the localization of the different molecular components, in response to changes in the environment. The big challenge of *functional genomics* or *systems biology* consists in relating these structural and functional data to each other, in order to arrive at a global interpretation of the functioning of the organism [3, 4]. This amounts to predicting and understanding how the observed behavior of the organism—the adaptation to its environment, the differentiation of its cells during development, even its evolution on a longer time scale—emerges from the networks of molecular interactions.

The molecular interactions in the cell are quite heterogeneous in nature. They concern the transcription and translation of a gene, the enzymatic conversion of a metabolite, the phosphorylation of a regulatory protein, and so forth. While studying a cellular process, it is often sufficient, at least to a first approximation, to focus on a part of the interaction network, dominated by a particular type of interaction. Thus biologists have become used to distinguish *metabolic networks*, *signal transduction networks*, *genetic regulatory networks*, and other types of network. Metabolic networks connect the small molecules of the cell by way of enzymatic reactions, whereas the connectivity of signal transduction networks is to a large extent determined by the posttranslational modification of proteins. Genetic regulatory networks mainly concern interactions between proteins and nucleic acids, controling the transcription and translation of genes.

In this chapter, we focus on genetic regulatory networks, which play an important role in the functioning and differentiation of cells. For instance, they allow the genetic program of a bacterium, the level of expression of its genes, to be adapted in response to an external stress. A large part of the experimental data available today, notably transcriptome data, concern genetic regulatory networks. Not withstanding the importance of these networks, one should bear in mind that they are integrated in the cell with other types of network, sometimes to the point that it may be difficult to distinguish the one from the other.

Besides high-throughput experimental methods, mathematical and computational approaches are indispensable for the analysis of genetic regulatory networks. Given the large number of components of most networks of biological interest, connected by positive and negative feedback loops, an intuitive comprehension of the dynamics of the system is often difficult, if not impossible, to obtain. *Mathematical modeling* supported by *computer tools* can contribute to the analysis of a regulatory network by allowing the biologist to focus on a restricted number of plausible hypotheses. The formulation of a mathematical model requires an explicit and nonambiguous description of the hypotheses being made on the regulatory mechanisms under study. Furthermore, its simulation by means of the model yields predictions on the behavior of the cell that can be verified experimentally.

An approach for analyzing the dynamics of genetic regulatory networks—based on the coordinated application of experimental, mathematical, statistical, and computational tools—is summarized in Figure 6.1. As a first step, one or several initial models are constructed from previous knowledge of the system and

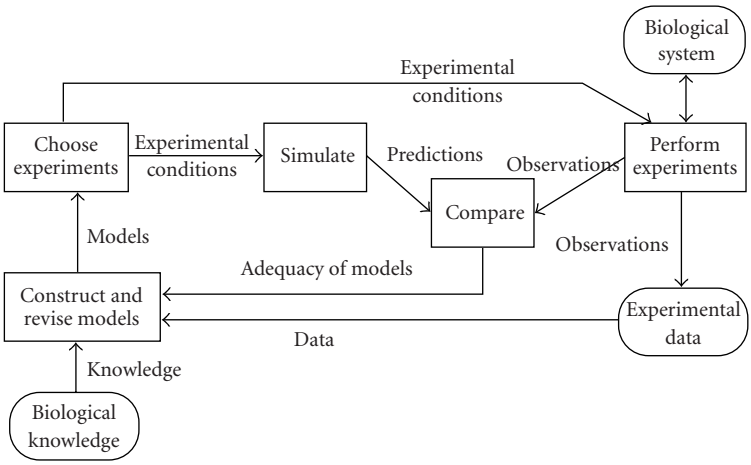


Figure 6.1. An approach for analyzing the dynamics of genetic regulatory networks: coordinated application of experimental, mathematical, statistical, and computational tools.

observations of its behavior. These models are then used to simulate the system under chosen experimental conditions, giving rise to predictions that can be compared with the observations made under the same experimental conditions. The fit between predictions and observations gives an indication of the adequacy of the models and may lead to their revision, thus initiating a new cycle of simulation of the behavior of the system, experimental verification of the predictions, and revision of the models.

In this chapter, we review methods for the modeling and simulation of genetic regulatory networks, showing how the approach summarized in Figure 6.1 can be put to work. In the last forty years, a large number of approaches for modeling genetic regulatory networks have been proposed in the literature, based on formalisms such as graphs, Boolean networks, differential equations, and stochastic master equations (Figure 6.2, see [5, 6, 7, 8, 9] for reviews). In their simplest form, *graph* models represent genetic regulatory networks by vertices (genes) connected by edges (interactions). More complicated graph models may label the edges with information on the type of interaction or associate probability distributions with the nodes, as in the case of Bayesian networks [10]. While graph models are static representations of genetic regulatory networks, *Boolean networks* describe their dynamics in a simple manner [11]. Using a Boolean variable for the state of a gene (on or off), and a Boolean function for the regulatory logic of the interactions, the temporal evolution of the state of the network can be described by means of a sequence of Boolean vectors. Various extensions of Boolean networks have been proposed in the literature, such as probabilistic Boolean networks [12] and generalized logical networks [13]. *Differential equations* provide a continuous, instead of discrete, description of the dynamics of genetic regulatory networks. The application of these models can be based on a well-established theoretical framework for modeling biochemical reactions [14, 15], while powerful analysis and simulation

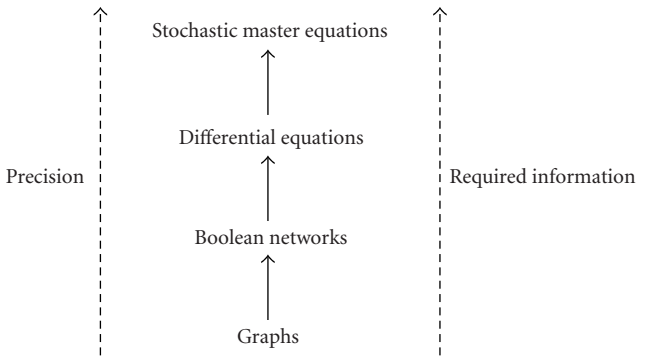


Figure 6.2. Hierarchy of different types of formalisms proposed for the modeling of genetic regulatory networks.

techniques exist. *Stochastic master equations* go beyond the deterministic character of differential equations, by recognizing that chemical reactions are basically stochastic processes [16, 17, 18]. The dynamics of genetic regulatory networks are modeled in the latter formalism as the temporal evolution of the probability distribution of the number of molecules of the different molecular species. The solution of the master equation can be approximated by means of stochastic simulation.

By climbing up the hierarchy from graph models to stochastic master equation models (Figure 6.2), we obtain increasingly precise descriptions of genetic regulatory networks. Not suprisingly, the information required for the application of the models to actual regulatory networks, as well as the necessary computational resources, increase in parallel, thus making it more difficult to use fine-grained models in practice. In this chapter, we restrict the discussion to *ordinary differential equations*, which probably form the most widely used class of models of genetic regulatory networks. However, as we argue in more detail later in this chapter, ordinary differential equation models should not be seen as appropriate for each and every problem. In the end, the suitability of a modeling formalism depends on the extent to which it is adapted to the biological problem under study.

In Section 6.2, we introduce some biological notions, fundamental for understanding the nature of genetic regulation and genetic regulatory networks. The next three sections focus on three different kinds of ordinary differential equation: nonlinear, linear, and piecewise-linear. In each section, we illustrate the properties of the kind of model with an example of a network of two genes, and we describe its application to real biological systems. The chapter ends with a more general discussion on the strengths and weaknesses of the modeling and simulation methods summarized here, taking into account the stochastic nature of biological processes and the complexity of the networks controling their evolution.

6.2. Genetic regulatory networks

We illustrate the main interactions in a genetic regulatory network by means of an example: the regulation of the expression of the sigma factor σ^S in *Escherichia coli*.

The role of the sigma subunits of the RNA polymerase is to recognize specific transcription initiation sites on the DNA, the so-called *promoters*. The expression or activation of a certain sigma factor therefore leads to the expression of a specific subset of genes of the organism. This type of regulation is often used by bacteria to assure a global response to an important change in their environment. Because of their importance for the global functioning of the cell, the expression of the sigma factors themselves is often tightly regulated.

E. coli possesses seven different sigma factors [19, 20]. The principal sigma factor, σ^{70}, directs the transcription of the so-called housekeeping genes. In many stress situations (lack of nutriments, high osmolarity, change of pH or of temperature, etc.), *E. coli* expresses the alternative sigma factor σ^S, encoded by gene *rpoS*. σ^S takes its name from the fact that it plays an important role in the adaptation to a particular stress, frequently encountered by bacteria: the depletion of nutriments in the environment, which leads to a considerable slowing down of cell growth, called the stationary growth phase. However, σ^S is activated in response to many other kinds of stress [21]. In order to illustrate the complexity of a genetic regulatory network, as well as the underlying molecular mechanisms, the following sections briefly summarize the different interactions by which the concentration of σ^S in the cell is regulated.

6.2.1. Regulation of transcription

Although regulation of transcription constitutes the preferred mode of regulating gene expression in bacteria, few studies have addressed this subject in the case of *rpoS*. As a consequence, our knowledge on the transcriptional regulation of this gene remains incomplete. Protein CRP, a typical repressor-activator, specifically binds the DNA at two sites close to the major promoter of *rpoS* [21]. One of these sites overlaps with the promoter, which implies that CRP and RNA polymerase cannot simultaneously bind to the DNA, due to sterical constraints. As a consequence, CRP represses the transcription of *rpoS*. The second binding site of CRP is located just upstream of the promoter. This geometry is reminiscent of the *lac* operon, where CRP binding to a similarly-positioned site establishes protein-protein interactions with the RNA polymerase, thus favoring the recruitment of RNA polymerase to the promoter [22]. The molecular details of this apparently contradictory regulation of the transcription of *rpoS* by CRP are still only partially understood. Nevertheless, the example illustrates one type of regulation that is quite widespread in bacteria: a protein binding the DNA (the regulator) prevents or favors the binding of RNA polymerase to the promoter.

A second factor regulates the transcription of *rpoS* as a function of the physiological state of the cell: when *E. coli* lacks amino acids, it synthesizes a small signaling molecule, guanosine tetraphosphate or ppGpp [23]. This molecule directly binds to the RNA polymerase and increases the activity of the latter at certain promoters—in particular, those involved in the biosynthesis of amino acids—and reduces the activity of RNA polymerase at other promoters—notably the ribosomal RNA promoters. A genetic analysis clearly shows that ppGpp strongly activates

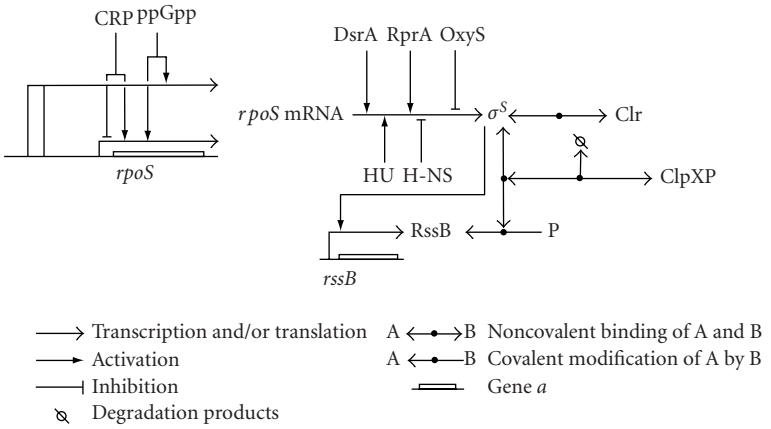


Figure 6.3. Regulation of the synthesis and degradation of the sigma factor σ^S. Only the interactions detailed in the text are shown. The notation is inspired by the graphical conventions proposed by Kohn [28].

the transcription of *rpoS*, but the same experiments also suggest that ppGpp does not act at the level of transcription initiation [24, 25]. The hypothesis currently favored is that ppGpp activates the transcription of *rpoS* indirectly, *via* the inhibition of exopolyphosphatase, the enzyme degrading polyphosphate in the cell [26]. However, the molecular mechanism of the activation of transcription of *rpoS* by polyphosphate remains badly understood.

The different interactions regulating the transcription of *rpoS*, as detailed in the above paragraphs, are summarized in Figure 6.3. Other factors are involved in the regulation of the expression of σ^S, we mention here protein BarA, which is part of a two-component system detecting several environmental parameters, such as iron starvation or the contact with the epithelium of the urinary tract during infection by pathogenic strain *E. coli* [27].

6.2.2. Regulation of translation

The expression of *rpoS* is regulated not only on the transcriptional level but also at the posttranscriptional level. The *translation* of the mRNA of *rpoS* is stimulated by environmental stress factors such as high osmolarity, low temperature, or low pH. The translation begins with the recognition of the so-called *Shine-Dalgarno sequence* by the ribosome, followed by the binding of the latter to this sequence. The efficiency of translation depends on the similarity of the Shine-Dalgarno sequence to the consensus sequence, and its accessibility to the ribosome. The mRNA of *rpoS*, like any other RNA, is not only a linear molecule, but possesses a particular secondary and tertiary structure. If the Shine-Dalgarno sequence is sequestered in a secondary structure (e.g., an RNA helix), it will be less accessible to the ribosome and the efficiency of translation will be reduced.

In the case of *rpoS*, at least three small regulatory RNAs and an equal number of proteins modify the structure of the RNA near the Shine-Dalgarno sequence.

In this way, they are able to modulate the efficiency of translation [21, 29]. By default, the mRNA of *rpoS* folds into a secondary structure that sequesters the Shine-Dalgarno sequence in a double helix. An alternative helix structure is formed in the presence of the small RNAs DsrA or RprA, thus releasing the binding site of the ribosome and increasing the efficiency of translation. Genes *dsrA* and *rprA* are transcribed in response to different environmental stresses: the expression of *dsrA* is strongly increased at low temperatures, while *rprA* transcription is regulated by stress factors modifying the surface properties of the cell (such as osmolarity or the presence of other bacteria). A third regulatory RNA, OxyS, inhibits the translation of *rpoS*. OxyS accumulates in the cell following an oxidative stress. The molecular mechanism of this regulation is not well understood, even though convincing evidence demonstrates the involvement of Hfq, a protein necessary for the rapid association of small RNAs with their targets.

Other proteins exert an influence on the translation of *rpoS*. We can cite at least two "histone-like" proteins, called HU and H-NS. HU binds with high affinity to the mRNA of *rpoS* and modifies the structure of the RNA [30]. The molecular details are not known, but the consequence of the binding of HU to the mRNA of *rpoS* is a strong increase in the translation of the latter. Protein H-NS has an inverse effect [21], but the molecular mechanism is not understood either, H-NS can bind to the mRNA of *rpoS*, but the specificity of the interaction has not been proven yet. It is equally possible that the effect of H-NS is indirect, for instance by preventing the action of a positive regulator like HU. Although the molecular details remain unclear, molecular genetics experiments have allowed to infer the regulatory logic. As we will see in the following sections, this knowledge is often sufficient to predict the global behavior of the network.

6.2.3. Regulation of degradation and activity

Not only the synthesis of σ^S is tightly regulated, but its *stability* and *activity* are also subject to multiple environmental and physiological influences. Like many proteins in *E. coli*, σ^S is degraded by proteases. All proteins are recognized by proteases when they are misfolded or truncated. In addition, certain proteins contain sequences (often at the N- or C-terminal region) that are specifically recognized by proteases. In the case of σ^S, a highly specialized system targets the protein for degradation by the ATP-dependent protease ClpXP. Protein RssB, when phosphorylated, forms a tight complex with σ^S [31]. RssB also interacts with ClpX, the subunit of the ClpXP complex that recognizes the substrates of the protease, and thus targets σ^S towards ClpXP. The catalytic subunit, ClpP, degrades σ^S and phosphorylated RssB is released, ready to dispatch another σ^S molecule towards degradation. The system is finely regulated by a feedback loop: the synthesis of RssB depends on σ^S [32].

In addition to this homeostatic regulation, the σ^S-RssB system is subject to environmental signals. RssB only binds to σ^S, if it is phosphorylated. RssB is a response regulator in a two-component system, a signal transduction mechanism frequently encountered in prokaryotes [33]. A so-called *sensor* protein detects

a physiological or an environmental signal and autophosphorylates itself on a histidine. The phosphate is then transferred to an aspartate of the second protein, called *response regulator* (in our case, RssB). The response regulator launches an appropriate response, such as the activation of a promoter or the degradation of a target protein (in our case, σ^S). The sensor corresponding to RssB has not been identified yet, so we do not know the signals modulating the phosphorylation of RssB and the capacity of the latter to target σ^S for degradation.

Even the stable protein σ^S is subject to further regulation. In order to fulfil its role as a recognition factor of promoters, σ^S must form a complex with the RNA polymerase. *E. coli* possesses seven different sigma factors that compete for binding to RNA polymerase. Certain proteins and small molecules influence this competition and thus control the "activity" of σ^S. It has been shown that ppGpp—in addition to its above-mentioned function in transcriptional initiation—favors the formation of the complex between σ^S and RNA polymerase [34]. We have recently discovered that the small protein Crl specifically interacts with σ^S, and increases the affinity of RNA polymerase containing σ^S for a subset of promoters recognized by this sigma factor (see http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=retrieve&db=pubmed&dopt=abstract&list_uids=14978043). This interaction connects the activity of σ^S to other physiological signals, due to the fact that Crl is regulated by temperature, growth phase, and other environmental stimuli.

6.2.4. Complexity of the network and interactions with other networks

Transcription, translation, and degradation are regulated by a large number of interactions, as illustrated here for σ^S. This convergence or "fan-in" of influences on *rpoS* and its protein is accompanied by an even more important divergence or "fan-out," in the sense that σ^S regulates the transcription of at least 70 genes of *E. coli* [21]. Among these genes, several encoding proteins that are directly or indirectly involved in the regulation of the synthesis and degradation of σ^S. This endows the network with a complex feedback structure, responsible for the adaptation of the transcriptional program of the bacterium to external perturbations.

The complexity of the genetic regulatory network is further increased by the fact that it is integrated with other networks. As we mentioned above, by citing the examples of the response regulators BarA and RssB, the σ^S regulon is the target of signals transduced by two-component systems. In the cases of BarA and RssB, we do not yet know the principal sensor of the signal transduction pathway. The σ^S regulon is also the target of regulatory factors originating in the cellular metabolism. In addition to the above-mentioned signaling molecule ppGpp, the expression of σ^S, or at least of genes dependent on σ^S, is also sensitive to the concentration of more classical metabolites, such as lactic acid or the redox state of the cell (as measured by the ratio of NADH and NAD^+). The presence of weak acids in the growth medium activates the expression of genes dependent on σ^S [35], whereas a high ratio of NADH and NAD^+ decreases the transcription of *rpoS* [36].

The genetic regulatory network controling the expression of *rpoS* as well as the regulation of expression of target genes of σ^S is thus embedded in the metabolic

and signal transduction networks of the cell. A complete understanding of the dynamics of this system would require a detailed description of all these elements. However, we can often abstract from the metabolic and signal transduction networks—by focusing on their effects on gene expression—and nevertheless obtain an adequate description of the global functioning of the regulatory system [37].

6.3. Nonlinear ordinary differential equation models

6.3.1. Equations and mathematical analysis

Nonlinear ordinary differential equations are probably the most-widespread formalism for modeling genetic regulatory networks. They represent the concentration of gene products—mRNAs or proteins—by continuous, time-dependent variables, that is, $x(t)$, $t \in T$, T being a closed time interval ($T \subseteq \mathbb{R}_{\geq 0}$). The variables take their values from the set of nonnegative real numbers ($x : T \to \mathbb{R}_{\geq 0}$), reflecting the constraint that a concentration cannot be negative. In order to model the regulatory interactions between genes, functional or differential relations are used.

More precisely, gene regulation is modeled by a system of ordinary differential equations having the following form:

$$\frac{dx_i}{dt} = f_i(\mathbf{x}), \quad i \in [1, \ldots, n], \tag{6.1}$$

where $\mathbf{x} = [x_1, \ldots, x_n]'$ is the vector of concentration variables of the system, and the function $f_i : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$, usually highly *nonlinear*, represents the regulatory interactions. The system of equations (6.1) describes how the temporal derivative of the concentration variables depends on the values of the concentration variables themselves. In order to simplify the notation, we can write (6.1) as the vector equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \tag{6.2}$$

with $\mathbf{f} = [f_1, \ldots, f_n]'$. Several variants of (6.2) can be imagined. For instance, by taking into account the input variables $\mathbf{u}$, it becomes possible to express the dependence of the temporal derivative on external factors, such as the presence of nutriments. In order to account for the delays resulting from the time it takes to complete transcription, translation, and the other stages of the synthesis and the transport of proteins, (6.2) has to be changed into a system of delay differential equations [9].

The above definitions can be illustrated by means of a simple network of two genes (Figure 6.4). Each of the genes encodes a regulatory protein that inhibits the expression of the other gene, by binding to a site overlapping the promoter of the gene. Simple as it is, this *mutual-inhibition network* is a basic component of more complex, real networks and allows the analysis of some characteristic aspects of cellular differentiation [13, 38].

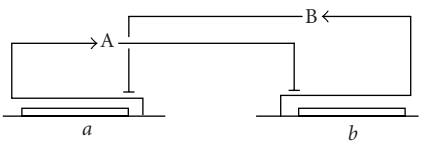
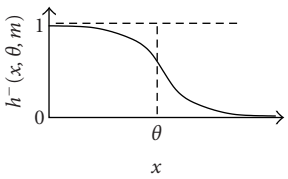


Figure 6.4. Example of a simple genetic regulatory network, composed of two genes a and b, the proteins A and B, and their regulatory interactions.

$$\frac{dx_a}{dt} = \kappa_a h^-(x_b, \theta_b, m_b) - \gamma_a x_a$$
$$\frac{dx_b}{dt} = \kappa_b h^-(x_a, \theta_a, m_a) - \gamma_b x_b$$
$$h^-(x, \theta, m) = \frac{\theta^m}{x^m + \theta^m}$$

(a)



(b)

Figure 6.5. (a) Nonlinear ordinary differential equation model of the mutual-inhibition network (Figure 6.4). The variables x_a and x_b correspond to the concentrations of proteins A and B, respectively, parameters κ_a and κ_b to the synthesis rates of the proteins, parameters γ_a and γ_b to the degradation constants, parameters θ_a and θ_b to the threshold concentrations, and parameters m_a and m_b to the degree of cooperativity of the interactions. All parameters are positive. (b) Graphical representation of the characteristic sigmoidal form, for $m > 1$, of the Hill function $h^-(x, \theta, m)$.

An ordinary differential equation model of the network is shown in Figure 6.5a. The variables x_a and x_b represent the concentration of proteins A and B, encoded by genes a and b, respectively. The temporal derivative of x_a is the difference between the *synthesis term* $\kappa_a h^-(x_b, \theta_b, m_b)$ and the *degradation term* $\gamma_a x_a$. The first term expresses that the rate of synthesis of protein A depends on the concentration of protein B and is described by the function $h^- : \mathbb{R}_{\geq 0} \times \mathbb{R}^2_{>0} \to \mathbb{R}_{\geq 0}$. This so-called *Hill function* is monotonically decreasing. It takes the value 1 for $x_b = 0$, and asymptotically reaches 0 for $x_b \to \infty$. It is characterized by a threshold parameter θ_b and a cooperativity parameter m_b (Figure 6.5b). For $m_b > 1$, the Hill function has a sigmoidal form that is often observed experimentally [39, 40]. The synthesis term $\kappa_a h^-(x_b, \theta_b, m_b)$ thus means that, for low concentrations of protein B, gene a is expressed at a rate close to its maximum rate κ_a ($\kappa_a > 0$), whereas for high concentrations of B, the expression of the gene is almost completely repressed. The second term of the differential equation, the degradation term, expresses that the degradation rate of protein A is proportional to its own concentration x_a, γ_a being a degradation parameter ($\gamma_a > 0$). In other words, the degradation of the protein is not regulated in this example. The differential equation for x_b has an analogous interpretation.

Because of the nonlinearity of the functions $\mathbf{f}$, the solutions of the system of ordinary differential equations (6.2) cannot generally be determined by analytical

means. This is even true for the nonlinear model of the two-gene network (Figure 6.5). However, because the model has only two variables, we can obtain a qualitative understanding of the dynamics of the network, by applying the tools available for analysis in the phase plane (see [41] for an accessible introduction).

The *phase plane* of the system is represented in Figure 6.6. Every point in the plane represents a pair of concentrations x_a and x_b. The solutions of the system of differential equations give rise to *trajectories* in the phase plane, as illustrated in Figure 6.6a. Another way of studying the dynamics of the system consists in analyzing the *vector field*, that is, the vector of temporal derivatives $[dx_a/dt, dx_b/dt]'$ associated with each point. This gives an indication of the direction of the trajectories passing through the point, as illustrated in Figure 6.6b. The analysis can be refined by tracing the *nullclines* in the phase plane, that is, the curves on which the temporal derivatives of x_a and x_b equal 0 (here, these curves are defined by $x_a = (\kappa_a/\gamma_a)h^-(x_b, \theta_b, m_b)$ and $x_b = (\kappa_b/\gamma_b)h^-(x_a, \theta_a, m_a)$. The points where the nullclines intersect are the *equilibrium points* of the system. If all trajectories in a neighborhood of the equilibrium point remain in that neighborhood, then the equilibrium point is *stable*. If, in addition, they converge towards the equilibrium point, the latter is *asymptotically stable*. So, by studying the vector field around the equilibrium point, one can determine its stability. In the case of the nonlinear model of the network in Figure 6.4, there are three equilibrium points: two of these are asymptotically stable and one is unstable (Figure 6.6). The result of the analysis summarized in this paragraph is often called the *phase portrait*.

The above phase-plane analysis predicts that the mutual-inhibition network is *bistable*. That is, starting from certain initial conditions, the system will reach one of the two stable equilibria. From a practical point of view, the unstable equilibrium has no importance, because it is only attained for quite specific initial conditions. Moreover, a perturbation of the unstable equilibrium, even vanishingly small, will cause the system to converge towards one of the stable equilibria. The phase portrait also reveals that the system exhibits hysteresis. If one perturbs the system from one of its stable equilibria—for instance, by provoking a strong degradation of the protein present at a high concentration—the other equilibrium can be reached (Figure 6.6c). From then onwards, even if the source of strong degradation has disappeared, the system will remain at the new equilibrium. In other words, the example suggests that a simple molecular mechanism may allow the system to switch from one functional mode to another. For this reason, mutual-inhibition networks, or more generally networks with positive feedback loops, have been assigned a central role in cellular differentiation [13].

It is important to remark that the above analysis is not just a theoretical exercise. In fact, the properties of the mutual-inhibition network revealed by the analysis—bistability and hysteresis—have been experimentally tested by Gardner et al. [42]. The network of Figure 6.4 has been reconstructed in *E. coli* cells by cloning the genes on a plasmid. The genes have been chosen such that the activity of the corresponding proteins can be regulated by external signals. In addition, reporter genes have been added that allow the state of the cell to be measured.

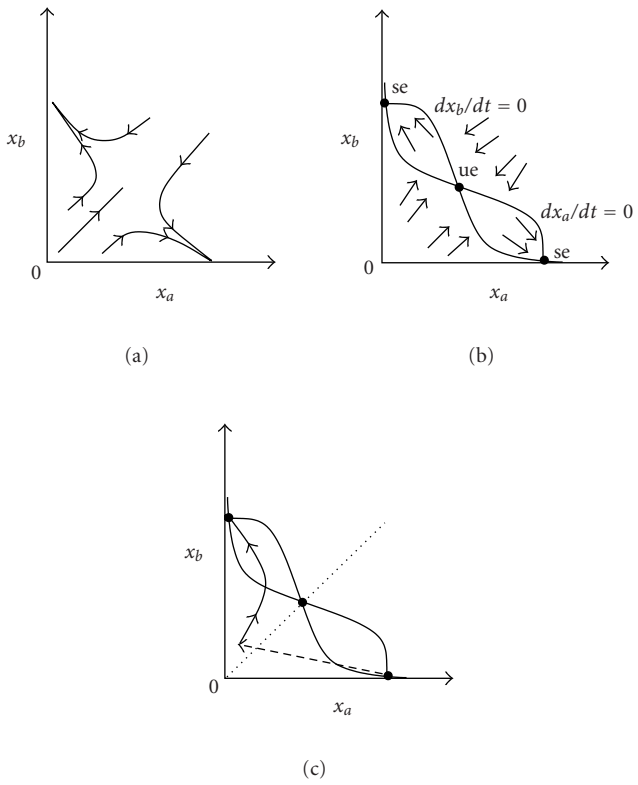


Figure 6.6. Phase portrait of the nonlinear model of the mutual-inhibition network (Figure 6.5). (a) Examples of trajectories. (b) Vector field and nullclines. The system has two asymptotically stable equilibrium points (se) and one unstable equilibrium point (ue). (c) Hysteresis effect resulting from a transient perturbation of the system (broken line with arrow).

The resulting mutual-inhibition network functions independently from the rest of the cell, like a "genetic applet," in the words of the authors. Carefully chosen experiments have shown that the system is bistable and can switch from one equilibrium to the other following chemical or heat induction.

The qualitative analysis of the dynamics of the mutual-inhibition network, summarized in Figure 6.6, is valid for a large range of parameter values. However, for certain parameter values, the behavior of the system changes, as can be verified in Figure 6.7. By increasing the value of parameter θ_b, the nullcline of x_a, defined by $x_a = (\kappa_a/\gamma_a)h^-(x_b, \theta_b, m_b)$, moves upwards. As a consequence, one of the stable equilibria and the unstable equilibrium approach and then annihilate each other. For values of θ_b close to, or above, κ_b/γ_b, the system loses its bistability and hysteresis properties. In the terminology of dynamical systems theory, a *bifurcation* has occurred [41].

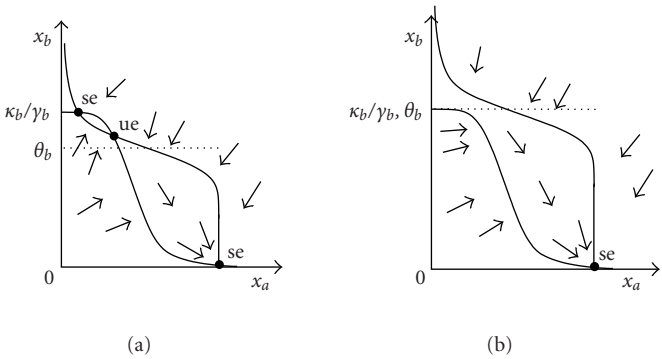


Figure 6.7. Analysis of the bifurcation occurring when the value of parameter θ_b is increased. The value in (a) is smaller than the value in (b).

Generally, for networks having more than two genes, an analysis in the phase plane is no longer possible. In certain cases, one can reduce the dimension of the system by simplifying the model, but most of the time, numerical techniques become necessary. *Numerical simulation* approximates the exact solution of the system of equations, by computing approximate values $\mathbf{x}_0, \ldots, \mathbf{x}_m$ for $\mathbf{x}$ at consecutive time points $t_0, \ldots, t_m$ (see [43] for an introduction). Many computer tools for numerical simulation have been developed, some specifically adapted to networks of molecular interactions. Well-known examples of the latter are GEPASI [44], DBsolve [45], and Ingeneue [46]. Recently, formats for the exchange of models between different simulation tools have appeared [47], as well as platforms enabling interactions between different tools [48].

Numerical simulation tools are at the heart of the analysis of nonlinear models of genetic regulatory networks. Unfortunately, their practical application is often difficult, due to the general absence of *in vitro* and *in vivo* measurements of the parameters of the model. These values are only available for a few systems whose functioning has already been well characterized experimentally. Several solutions exist for dealing with the lack of quantitative data on the network components and their interactions. A first approach consists in using the increasing amounts of expression data, obtained by, for example, DNA microarrays or quantitative RT-PCR (reverse transcriptase-polymerase chain reaction). Starting with measurements of the concentration variables $\mathbf{x}$ at several stages of the process under investigation in different experimental conditions, the parameter values can be estimated by means of *system identification* techniques [49]. This approach will be examined in more detail in Section 6.4. Here, we focus on another solution to the problem of the lack of measured values of the kinetic parameters. This approach, illustrated by means of a study of the establishment of segment polarity in *Drosophila melanogaster*, is based on the hypothesis that essential properties of the system are robust to variations in parameter values.

6.3.2. Simulation of the establishment of segment polarity in *D. melanogaster*

The first stages of the development of the embryo of the fruit fly *D. melanogaster* consist of a segmentation of the anterio-posterior axis by the maternal genes and the *gap*, *pair-rule*, and *segment-polarity* gene classes [50, 51]. The majority of segmentation events take place before cellularization, in a syncytium in which transcription factors can freely diffuse. This leads to the establishment of concentration gradients that control the expression of genes playing a role later in development. After cellularization has occurred, the anterio-posterior axis is divided into 15 regions, prefiguring the segments of the embryo.

The identity of the segments is determined by the expression of the segment-polarity genes, of which the principal ones are *wingless (wg)*, *hedgehog (hh)*, and *engrailed (en)*. These genes code for transcription factors that are secreted and can therefore modulate gene expression in neighboring cells. The initial activation of the segment-polarity genes is largely determined by the pair-rule genes but, once established, the spatiotemporal expression profile of these genes is stable—until the fly has reached adulthood—and entirely determined by the interactions between the segment-polarity genes. This expression profile is expected to be quite robust, because it identifies the future function of each segment of the embryo.

The segmentation process has been studied in sufficient detail to allow us to draw an outline of the genetic regulatory network connecting the segment-polarity genes. In a recent publication, von Dassow et al. [52] propose such a network including the three genes mentioned above, as well as two other genes: *cubitus interruptus* and *patched* (Figure 6.8). Quite a few interactions between these genes are known, which makes it possible to formulate an ordinary differential equation model similar to the models presented in Section 6.3.1. However, in order to reproduce the expression profiles, the model has to take into account not only the temporal, but also the spatial aspects of the establishment of segment polarity. The model proposed by the authors includes a row of cells along the anterio-posterior axis, such that each of the cells contains a copy of the genetic regulatory network and interacts with its neighboring cells.

The resulting model contains almost 50 parameters, the values of which are unknown in most cases. The authors therefore searched for a set of parameters that would lead, starting from reasonable initial conditions (determined by the pair-rule genes), to a behavior of the network that is consistent with the biological observations, that is, corresponding to the observed expression profile in the embryo. Despite an extensive search, no such set of parameters could be found (Figure 6.9b). By analyzing the reasons for this negative result, the authors concluded that there must be missing interactions. In fact, when they added two additional interactions, suggested by genetic experiments and circumstantial evidence, a large number of parameter sets led to the desired behavior (Figure 6.9c). In fact, the revised model shows an extraordinary stability to variations in the parameter values, sometimes spanning several orders of magnitude.

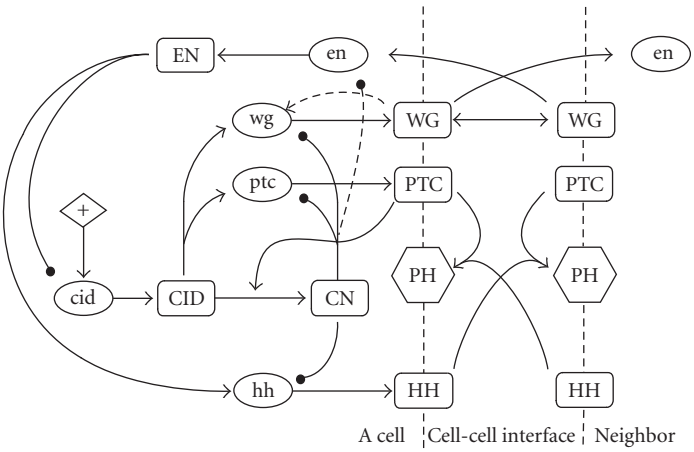


Figure 6.8. Interactions between the products of the five genes included in the segment-polarity model of von Dassow et al. (figure reproduced from [52]). The following abbreviations have been used: WG (wingless), EN (engrailed), HH (hedgehog), CID (C. interruptus), CN (repressor fragment of C. interruptus), PTC (patched), and PH (complex of patched and hedgehog). The figure represents the RNAs (ellipses) as well as the proteins (rectangles), while the positive interactions are indicated by arrows and the negative interactions by circles. The rhombus containing the plus sign indicates that *C. interruptus* has a basal expression level. The interactions denoted by a broken line were added after the other interactions had been shown to be insufficient to reproduce the observed expression profile (see the text).

This surprising result suggests that it may not be the exact value of the parameters that accounts for the functioning of the system, but rather the structure of the regulatory network. One might of course argue that the robustness of the developmental dynamics is particular to the segment-polarity network, which counts among the most important in *Drosophila*. As we mentioned above, the expression profile of the segment-polarity genes directs the further development of the organism. However, the robustness of properties decisive for the functioning of a network of molecular interactions has been observed in other cases as well. Large variations in parameter values do not change, for instance, the behavior of the neurogenic network in *Drosophila* [53] and the chemotaxic response in *E. coli* [54].

The segment-polarity example illustrates several construction principles of genetic regulatory networks. First, crucial properties of the system probably have to be robust against variations in the parameter values, due to inevitable mutations of the genes. As a consequence, the structure of the network is largely responsible for the behavior. Second, modeling can be used to study the dynamical consequences of the connectivity of the genetic regulatory network. Also, as illustrated in the example, modeling can suggest missing interactions that can then be verified experimentally. Finally, the fact that the behavior of the organism emerges from the interactions between a limited number of genes might be an indication that biological networks are organized in a modular fashion [55]. The multiple

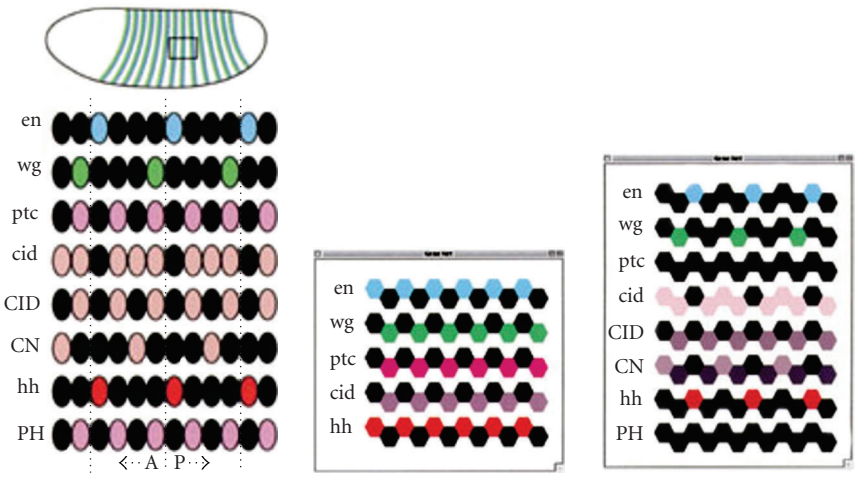


Figure 6.9. Spatial expression profiles of the segment-polarity genes in the cells along the anterio-posterior axis: (a) observed profile, (b) profile predicted by the model without additional interactions, (c) idem, but with additional interactions (figures reproduced from [52]).

additional influences on these genes—which may modulate the strengths of their interactions, as reflected by the parameter values—do not affect the fundamental dynamics of the network.

6.4. Linear ordinary differential equation models

6.4.1. Equations and mathematical analysis

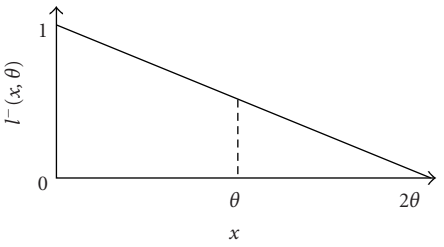
Nonlinear ordinary differential equation models give an adequate description of important aspects of the dynamics of genetic regulatory networks, as shown by the *in vivo* construction of the mutual-inhibition network by Gardner et al. (Section 6.3). Unfortunately, the nonlinear models become quite difficult to treat mathematically when passing from simple synthetic networks to the complex networks involved in, for example, the control of the establishment of segment polarity in *Drosophila*. This raises the question whether the dynamics of genetic regulatory networks could be equally well described by linear differential equation models, which possess more favorable mathematical properties.

A system of linear ordinary differential equations has the form (6.2), but the functions $\mathbf{f}$ are *linear*. That is, (6.2) can be rewritten as follows:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{n\times n}, \ \mathbf{b} \in \mathbb{R}^n. \tag{6.3}$$

$$\frac{dx_a}{dt} = \kappa_a l^-(x_b, \theta_b) - \gamma_a x_a$$
$$\frac{dx_b}{dt} = \kappa_b l^-(x_a, \theta_a) - \gamma_b x_b$$
$$l^-(x, \theta) = 1 - \frac{x}{2\theta}$$

(a)



(b)

$$\frac{d}{dt}\begin{bmatrix} x_a \\ x_b \end{bmatrix} = \begin{bmatrix} -\gamma_a & -\frac{\kappa_a}{(2\theta_b)} \\ -\frac{\kappa_b}{(2\theta_a)} & -\gamma_b \end{bmatrix}\begin{bmatrix} x_a \\ x_b \end{bmatrix} + \begin{bmatrix} \kappa_a \\ \kappa_b \end{bmatrix}$$

(c)

Figure 6.10. (a) Linear ordinary differential equation model of the mutual-inhibition network (Figure 6.4). The variables x_a and x_b correspond to the concentrations of proteins A and B, respectively, parameters κ_a and κ_b to the synthesis rates of the proteins, parameters γ_a and γ_b to the degradation constants, and parameters θ_a and θ_b to the strength of the interactions. All parameters are positive. (b) Graphical representation of the linear function $l^-(x, \theta)$. (c) Reformulation of the model in (a) in the matrix form of (6.3).

Henceforward, we make the hypothesis that the element of the matrix **A** and the vector **b** are constants. As a consequence, system (6.3) has an analytical solution, given by linear systems theory [56].

How can we model a genetic regulatory network by means of linear ordinary differential equations? By way of example, the model of the mutual-inhibition network is shown in Figure 6.10a. It much resembles the nonlinear model presented in Section 6.3.1: as before, the time derivative is equal to the difference between a synthesis term and a degradation term. However, a linear function $l^- : D \times \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$, $D \subset \mathbb{R}_{\geq 0}$ is now used instead of the sigmoidal function h^-. As the latter function, l^- is monotonically decreasing, but it is characterized by a single parameter, θ, defining the slope. In addition, the domain of the variable x is restricted to $D = [0, 2\theta] \subset \mathbb{R}_{\geq 0}$, because $1 - x/(2\theta)$ becomes negative for $x > 2\theta$, thus violating the obvious constraint that the synthesis rate must be nonnegative (Figure 6.10b). It is easily verified that the model can be rewritten in the form (6.3) (Figure 6.10c). Note that the model is only valid for $x_a \in [0, 2\theta_a]$ and $x_b \in [0, 2\theta_b]$, due to the definition of l^-.

As in the case of the nonlinear model, the qualitative dynamics of the network can be studied in the phase plane. Figure 6.11a shows some examples of trajectories in the phase plane. From a superficial comparison with Figure 6.6a, one

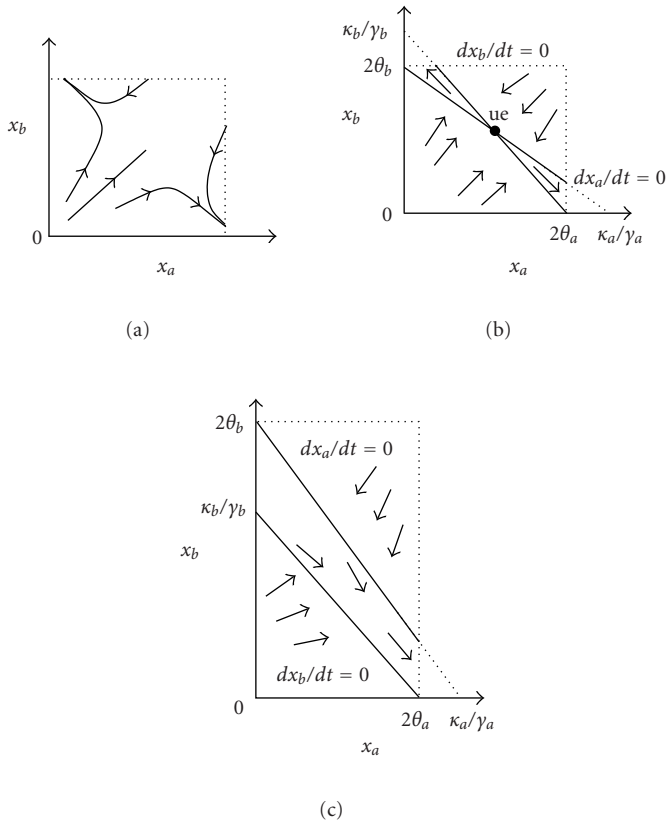


Figure 6.11. Phase portrait of the linear model of the mutual-inhibition network (Figure 6.10). (a) Examples of trajectories. (b) Vector field and nullclines. The system has a single unstable equilibrium point (ue). (c) Analysis of the bifurcation occurring when the value of parameter θ_b is increased. The value of θ_b in (c) is larger than that in (b). The analysis is restricted to $[0, 2\theta_a] \times [0, 2\theta_b]$, the part of the phase space where the linear model is defined.

would be inclined to conclude that the linear and nonlinear models make more or less identical predictions of the dynamics of the system. However, analysis of the nullclines—defined by $x_a = (\kappa_a/\gamma_a)l^-(x_b, \theta_b)$ and $x_b = (\kappa_b/\gamma_b)l^-(x_a, \theta_a)$—shows that this is not the case (Figure 6.11b). In fact, the linear system has only a single equilibrium point corresponding to the unstable equilibrium point of the nonlinear system in Figure 6.6b. Almost all trajectories reach one of the segments $x_a = 2\theta_a$ or $x_b = 2\theta_b$ after a while and would continue towards $(-\infty, \infty)'$ or $(\infty, -\infty)'$, respectively, if the system were defined outside $[0, 2\theta_a] \times [0, 2\theta_b]$. Figure 6.11c shows that the equilibrium point disappears if one increases the value $2\theta_b$ above κ_b/γ_b, while keeping the other parameters at the same value. In that case, all trajectories reach the segment $x_a = 2\theta_a$.

The phase-plane analysis summarized in Figure 6.11 teaches us that, when modeled by a system of linear differential equations, the mutual-inhibition network no longer exhibits bistability or hysteresis. The predictions of the model therefore contradict what is experimentally observed by Gardner et al. In fact, the example shows that the nonlinear character of the inhibition of gene expression by regulatory proteins, expressed by means of the function h^-, is crucial for the global dynamics of the network. The approximation of h^- by l^- is unable to preserve essential properties of the dynamics. On the other hand, the analysis of the two-gene network suggests that linear models could contribute to the analysis of the local dynamics of the system. For example, even though they do not converge towards a stable equilibrium point, the trajectories in Figure 6.11a resemble those predicted by the nonlinear model in the neighborhood of the unstable equilibrium (Figure 6.6a).

This property of linear models can be exploited when trying to reconstruct the connectivity of a genetic regulatory network from experimental data. Suppose one had a time series of measurements of the concentration variables, obtained by DNA microarrays or quantitative RT-PCR. This series of measurements can be represented in the form of a matrix $\hat{\mathbf{X}}$, where $\hat{\mathbf{X}} \in \mathbb{R}^{n \times m}$. Every element $\hat{x}_{ij}$ of this matrix represents a measurement, more specifically the measurement of the variable x_i at time point j. Instead of a time series of measurements, the columns of the matrix $\hat{\mathbf{X}}$ could also represent measurements realized under various experimental conditions, for instance, the steady state reached by a mutant of the organism after a physiological perturbation.

System identification techniques [49] allow the values of the elements of the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ to be estimated from measurements $\hat{\mathbf{X}}$. These estimations make it possible to infer the interaction structure of a network, as can be easily understood by considering the matrix $\mathbf{A}$ in the case of the mutual-inhibition network (Figure 6.10c, see also [37, 57]). In fact, the negative sign of the off-diagonal elements a_{ab} and a_{ba} corresponds to the inhibition of gene a by protein B, and the inhibition of gene b by protein A, respectively. If B activated a and A activated b, these elements would have been positive (which can be simply verified by replacing $l^-(x, \theta)$ by $l^+(x, \theta) = 1 - l^-(x, \theta)$ in the model of Figure 6.10a). More generally, it follows that the estimation of the values of $\mathbf{A}$ and $\mathbf{b}$ from expression data provides us with information about the regulatory structure of the system, that is, on the existence of interactions between the genes and the nature of these interactions (activation, inhibition).

From a technical point of view, the use of linear ordinary differential equation models simplifies the approach of reconstructing genetic regulatory networks from gene expression data. In comparison with nonlinear models, linear models have a restricted number of parameters, as can be seen by comparing the models in Figures 6.5a and 6.10a. In addition, powerful techniques for parameter estimation exist for linear models. Taken together, this makes linear models more adapted to the quantitative and qualitative limitations of the experimental data available today. In fact, the expression data obtained by DNA microarrays are often noisy and the number of measurements m much smaller than the number of variables n.

Therefore, most studies on network reconstruction to date have used linear or pseudolinear models. In the next section, we discuss one recent example in more detail.

6.4.2. Reconstruction of the SOS regulon in *E. coli*

A large number of environmental stresses may damage the DNA. The bacterium responds to these attacks by expressing a certain number of genes allowing the DNA to be repaired or to be replicated despite it being damaged. In *E. coli* and other bacteria, the regulatory system coordinating the response to this type of stress is called the *SOS regulon* [58, 59]. The central regulator is not a sigma factor, as in the case of the general stress response described in Section 6.2, but involves the interactions between two proteins: RecA and LexA.

LexA is a transcription factor repressing all genes of the SOS regulon during normal growth by binding to operator sites overlapping the promoters of its target genes. Protein LexA is a dimer and each monomer is composed of two domains. Besides its binding activity to DNA, LexA has the ability to cut the connection between the two domains of each monomer. The affinity of LexA for its DNA binding sites strongly diminishes after having been cut into two, as a consequence of which the genes of the SOS regulon are derepressed. However, the autoproteolytic activity of LexA remains quite weak when it does not interact with RecA in a particular conformational state.

The main function of RecA, an important protein for which homologues exist in all organisms, is to catalyze DNA strand exchanges during homologous recombination. RecA possesses a high affinity for single-stranded DNA. This form of DNA is an intermediate of the recombination reaction, but DNA damage also, directly or indirectly, leads to its appearance in the cell. The formation of a complex with single-stranded DNA induces RecA to change its conformation, which allows it to interact with LexA and to stimulate the autoproteolytic activity of the latter.

In a certain way, RecA thus functions as a detector of DNA damage, while LexA regulates the response to this stress. The structure of the SOS regulon is therefore relatively simple. However, to this simple picture, we need to add the transcriptional autoregulation of LexA, the transcription of various genes of the regulon by three sigma factors (including σ^S), and the fact that key target genes allow the DNA to be repaired or replicated (Figure 6.12). The repair of the DNA removes the stimulus, so that LexA starts to accumulate in the cell again, the SOS regulon is repressed, and the normal growth state restored. By its relative simplicity, and by the fact that this regulon has been much studied, the SOS system is an excellent test case for two tasks: (i) inferring the connectivity of the genetic regulatory network from expression data, and (ii) characterizing the interactions through the estimation of kinetic parameters.

In [60], the group of Collins at Boston University tackled the first task, inferring the connectivity of the network from gene expression data. The authors chose nine key genes of the SOS system in *E. coli* and they measured, by quantitative RT-PCR, the RNA concentration of these genes at an equilibrium growth state.

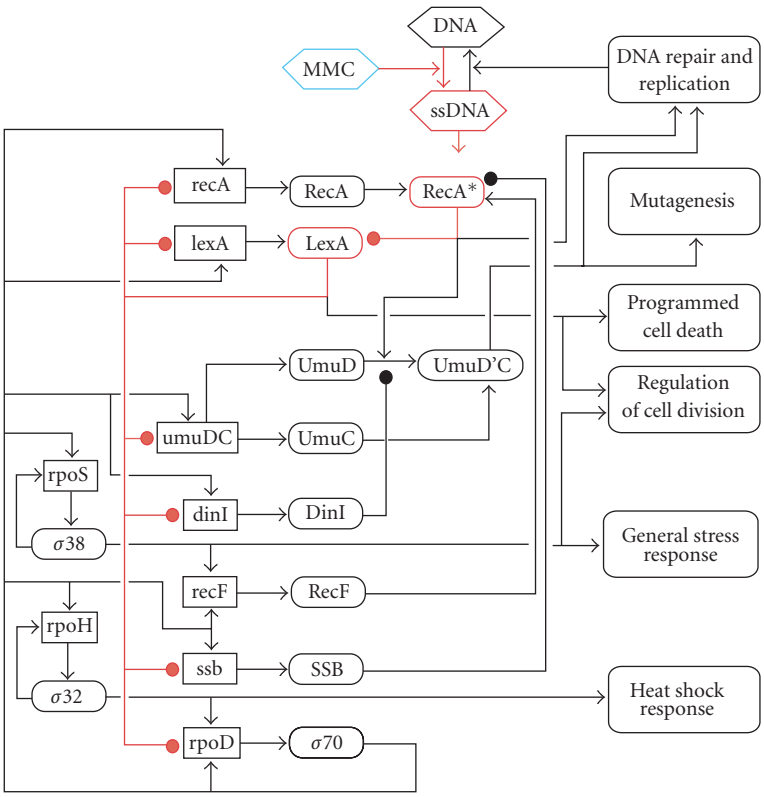


Figure 6.12. Genetic regulatory network of the SOS regulon in *E. coli* (figure reproduced from [60]). The figure represents genes (rectangles) as well as proteins (ellipses). Positive interactions are indicated by arrows and negative interactions by circles.

Next, they perturbed the regulatory network by overexpressing one of the nine genes. A new equilibrium was established after a transition period and the RNA concentrations were measured again. A considerable number of repetitions of the experiment and careful experimental work have thus led to reliable measurements of the expression level of the nine genes under nine different conditions (overexpression of each of the nine genes).

Close to the equilibrium, a linear description of the system often provides a good approximation of its dynamics. The regulatory interactions between the nine genes are therefore modeled by a system of linear differential equations of the form (6.3), where **b** represents external perturbations applied to the cell. In order to determine the parameters of the linear system, it would in principle be sufficient to measure the RNA concentration at nine different equilibrium states [49]. However, even careful measurements of the expression level are still too noisy to make this direct approach possible. For this reason, multiple regression is used to find the set of values for the parameters that fit the expression data with the

smallest error. In addition, the authors make the reasonable assumption that the regulatory network is only sparsely connected, that is, that each gene is regulated by few of the other genes. This restriction considerably reduces the number of possibilities to be explored.

The approach summarized above makes it possible to identify the major interactions of the regulatory network. For instance, the analysis correctly identifies the mutual inhibition of LexA and RecA, as well as the autoregulation of LexA. The model also correctly predicts that RecA and LexA exert the strongest regulatory influences in the SOS system. By analyzing the nine data sets, the authors uncover more than half of the known regulatory interactions between the nine genes of the network. However, about 40% of the predicted interactions by the linear analysis are false positives, that is, they do not occur in the real network. This is not surprising, if one takes into account the error level inherent in measurements of RNA concentrations. By computing the mean of sixteen measurements of each transcript (i.e., eight replications of each experiment, analyzed by two separate PCR reactions), the authors are able to measure the RNA concentration with an error (the ratio of the mean standard error and the mean) of about 68%. The error in an experiment using DNA microarrays is generally even larger. The authors show by means of simulations that a 10% reduction of the experimental error allows one to find 75% of the existing interactions with only 20% of false positives.

The example shows that an analysis by means of linear models can provide a good first sketch of an unknown regulatory network. The interpretation of the results must take into account not only that the network is probably incomplete and that false positives may be present, but also that the identified regulatory interactions are not necessarily direct. An enzyme increasing the concentration of a metabolite, which activates a kinase phosphorylating a transcription factor that activates a gene, would give rise to the activation of the gene by the enzyme [37]. An additional problem, inherent to linear models, is the difficulty to find cooperative interactions, involving several components of the network. For instance, the initiation of the transcription of a gene might require two transcription factors to simultaneously bind upstream of a promoter. The majority of interactions in the SOS regulon are direct and noncooperative, which allows the method based on linear models to provide a good idea of the connectivity of the network.

However, in order to better characterize the interactions of the network, going beyond the mere connectivity of the network, one has to pass from linear to nonlinear models. This allows the interactions to be described in more detail, but necessitates the use of more powerful methods and larger amounts of more precise data. The group of Alon at the Weizmann Institute of Science has recently taken up this challenge for the SOS regulon [61]. Given the connectivity of the network, somewhat simplified in comparison with Figure 6.12, the authors have estimated the values of the parameters characterizing the interactions.

The experimental approach of the authors is similar to the one used by the group of Collins for the reconstruction of the connectivity of the network. The expression of the genes being studied is not measured by quantitative RT-PCR,

but by transcriptional fusions of the promoters with the GFP (green fluorescent protein) gene. This allows the expression of a gene to be measured *in vivo* in real time. The high sampling frequency (one measurement every few minutes) makes it possible to follow the kinetics of the change in gene expression after a perturbation. A single perturbation is used in these experiments: irradiation with UV light, which entails DNA damage and induces the SOS system. Contrary to the experiments of the group of Collins, where the expression of the genes is determined at two successive equilibria, what is measured here is the transient phase between the perturbation of the equilibrium by means of UV light and the return to the same equilibrium after repair of the damage. Each expression profile consists of a time series of some 30 measurements, performed at three-minute intervals.

The dynamics of the SOS system is essentially described by eight nonlinear ordinary differential equations which describe the repression of eight target genes by LexA. Each equation is of the Michaelis-Menten form and therefore described by two parameters: the first represents the strength of the promoter and the second the apparent affinity of LexA for this promoter. The challenge consist in estimating the sixteen parameters from experimental data. After transformation of the model in a quasi-linear form, a standard optimization algorithm allows a set of parameter values to be found that minimizes the difference between predictions and observations. If the error in the prediction of the expression profile of a gene is too important, then one may suppose that a nonidentified interaction has been omitted. In the results presented by the authors, the error for gene *uvrY* is close to 45%. In fact, recent biological data suggest that this gene is subject to an additional regulation [62, 63].

Because it is assumed here that all regulatory interactions of the network are known, and the form of the differential equations is simple, one can estimate the parameter values from high-quality temporal expression profiles (the mean error is of the order of 10%). However, for most genetic regulatory networks these conditions are more difficult to satisfy than for the relatively simple and well-studied SOS regulon.

6.5. Piecewise-linear ordinary differential equation models

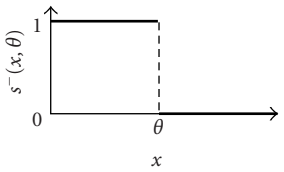
6.5.1. Equations and mathematical analysis

The linear differential equation models are easier to analyze than the nonlinear models, as we have seen in Section 6.4.1. However, this mathematical simplicity comes at the price of a reduced ability to take into account essential properties of the dynamics of the system. Are there models that are easy to treat mathematically and nevertheless capable of adequately representing the dynamics of the system? In this section, we study a class of models that answer both requirements: piecewise-linear differential equations.

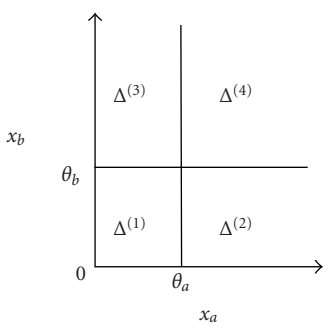
The general form of the models is given by (6.2), with the additional constraint that the functions $\mathbf{f}$ are *piecewise linear*. That is, the phase space $\mathbb{R}^n_{\geq 0}$ is

$$\frac{dx_a}{dt} = \kappa_a s^-(x_b, \theta_b) - \gamma_a x_a$$
$$\frac{dx_b}{dt} = \kappa_b s^-(x_a, \theta_a) - \gamma_b x_b$$
$$s^-(x, \theta) = \begin{cases} 1, & x < 0, \\ 0, & x > 0 \end{cases}$$

(a)



(b)



$$\Delta^{(1)}: \frac{d}{dt}\begin{bmatrix} x_a \\ x_b \end{bmatrix} = \begin{bmatrix} -\gamma_a & 0 \\ 0 & -\gamma_b \end{bmatrix}\begin{bmatrix} x_a \\ x_b \end{bmatrix} + \begin{bmatrix} \kappa_a \\ \kappa_b \end{bmatrix}$$
$$\Delta^{(2)}: \frac{d}{dt}\begin{bmatrix} x_a \\ x_b \end{bmatrix} = \begin{bmatrix} -\gamma_a & 0 \\ 0 & -\gamma_b \end{bmatrix}\begin{bmatrix} x_a \\ x_b \end{bmatrix} + \begin{bmatrix} \kappa_a \\ 0 \end{bmatrix}$$
$$\Delta^{(3)}: \frac{d}{dt}\begin{bmatrix} x_a \\ x_b \end{bmatrix} = \begin{bmatrix} -\gamma_a & 0 \\ 0 & -\gamma_b \end{bmatrix}\begin{bmatrix} x_a \\ x_b \end{bmatrix} + \begin{bmatrix} 0 \\ \kappa_b \end{bmatrix}$$
$$\Delta^{(4)}: \frac{d}{dt}\begin{bmatrix} x_a \\ x_b \end{bmatrix} = \begin{bmatrix} -\gamma_a & 0 \\ 0 & -\gamma_b \end{bmatrix}\begin{bmatrix} x_a \\ x_b \end{bmatrix}$$

(c)

Figure 6.13. (a) Piecewise-linear differential equation model of the mutual-inhibition network (Figure 6.4). The variables x_a and x_b represent the concentration of proteins A and B, respectively, parameters κ_a and κ_b the synthesis rates, parameters γ_a and γ_b the degradation constants, and parameters θ_a and θ_b threshold concentrations for A and B, respectively. All parameters are positive. (b) Graphical representation of the step function $s^-(x, \theta)$. (c) Piecewise-linear structure of the model in (a), corresponding to the division of the phase space into four regions $(\Delta^{(1)}, \ldots, \Delta^{(4)})$ by $x_a = \theta_a$ and $x_b = \theta_b$.

divided into regions $\Delta^{(j)}$, $j \in [1, \ldots, p]$, in each of which the network is described by a system of linear differential equations. While being *globally* nonlinear, a piecewise-linear differential equation model is *locally* linear

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}^{(j)}\mathbf{x} + \mathbf{b}^{(j)}, \quad \mathbf{A}^{(j)} \in \mathbb{R}^{n\times n},\ \mathbf{b}^{(j)} \in \mathbb{R}^n,\ \mathbf{x} \in \Delta^{(j)} \subseteq \mathbb{R}^n_{\geq 0},\ j \in [1, \ldots, p]. \tag{6.4}$$

As for the linear models, we assume that the elements of $\mathbf{A}^{(j)}$ and $\mathbf{b}^{(j)}$ are constants. This implies that in each region $\Delta^{(j)}$, (6.4) can be solved analytically.

Piecewise-linear differential equations have been used to model genetic regulatory networks since the early seventies [64, 65, 66, 67]. In order to illustrate their application, we again consider the example of the two-gene network. The piecewise-linear model, presented in Figure 6.13, is obtained from the nonlinear model by replacing the sigmoidal function h^- by another approximation, the step

function $s^- : D \times \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$, $D \subset \mathbb{R}_{\geq 0}$. For concentrations x below the threshold θ, $s^-(x, \theta)$ equals 1, whereas for concentrations x above θ, the function evaluates to 0. For $x = \theta$, it is not defined. As one can verify in Figure 6.13c, the model can be rewritten in the form (6.4). The segments $x_a = \theta_a$ and $x_b = \theta_b$ divide the phase space into four regions, $\Delta^{(1)}, \ldots, \Delta^{(4)}$. In each region, after evaluation of the step functions, the model reduces to a system of two linear differential equations.

In the cases that interest us, the reduced system of differential equations associated with a region $\Delta^{(j)}$ is not only linear, but also *uncoupled*. That is, $\mathbf{A}^{(j)}$ is a diagonal matrix and the temporal derivative of the variable x_i does not depend on variables other than x_i. Such a system has a very simple analytical solution. In fact, one can show that, in region $\Delta^{(j)}$, all solutions locally converge towards the point $\phi(\Delta^{(j)}) = (b_1^{(j)}/a_{11}^{(j)}, \ldots, b_n^{(j)}/a_{nn}^{(j)})'$ [64]. For instance, in $\Delta^{(1)}$, the solutions converge towards $\phi(\Delta^{(1)}) = (\kappa_a/\gamma_a, \kappa_b/\gamma_b)'$, while in $\Delta^{(2)}$, they converge towards $\phi(\Delta^{(2)}) = (\kappa_a/\gamma_a, 0)'$ (Figure 6.14b). If $\phi(\Delta^{(j)}) \in \Delta^{(j)}$, then $\phi(\Delta^{(j)})$ is an equilibrium point of the system, which is, for instance, the case for $\Delta^{(2)}$ and $\Delta^{(3)}$.

The piecewise-linear model does not specify how the system behaves on the segments $x_a = \theta_a$ and $x_b = \theta_b$, where one or more step functions, and hence the corresponding differential equations, are not defined. In order to treat this problem, Gouzé and Sari [68] have proposed an approach which consists of extending the differential equation model (6.4) to a differential inclusion model, following ideas developed in control theory. This solution exploits mathematical concepts that are outside the scope of this chapter, but for our purposes, it is sufficient to know that the approach is elegant from a theoretical point of view and easy to use in practice. In the example, it allows the local analysis of the dynamics of the network to be extended to regions of the phase space located on the segments $x_a = \theta_a$ and $x_b = \theta_b$, that is, $\Delta^{(5)}, \ldots, \Delta^{(9)}$ (Figure 6.14c). The results of the analysis are intuitive: the solutions of the system instantaneously traverse $\Delta^{(5)}, \ldots, \Delta^{(8)}$, whereas solutions reaching $\Delta^{(9)}$ can remain indefinitely in this region or leave it after a certain time (Figure 6.14d).

The local analyses of the dynamics of the system in the different regions of the phase space can be combined into a global analysis, as illustrated in Figure 6.15. The predictions of the piecewise-linear model are qualitatively equivalent to those obtained by the nonlinear model. The network has three equilibrium points, of which two are stable and one is unstable (Figure 6.15a). Figure 6.15c shows that a transient perturbation may cause the system to switch from one stable equilibrium to the other. As for the nonlinear model, an increase of the value of parameter θ_b, without changing the value of the other parameters, can bring about a bifurcation: one of the two stable equilibria and the unstable equilibrium disappear (Figure 6.15b). In summary, the example shows that, while facilitating the mathematical analysis, the piecewise-linear models allow us to preserve essential properties of the mutual-inhibition network. There are good reasons to believe that this is also true for other, more complex networks, but this has not been formally proven yet.

The analysis of the piecewise-linear model of the two-gene network suggests a discrete, more compact representation of the dynamics of the system [69].

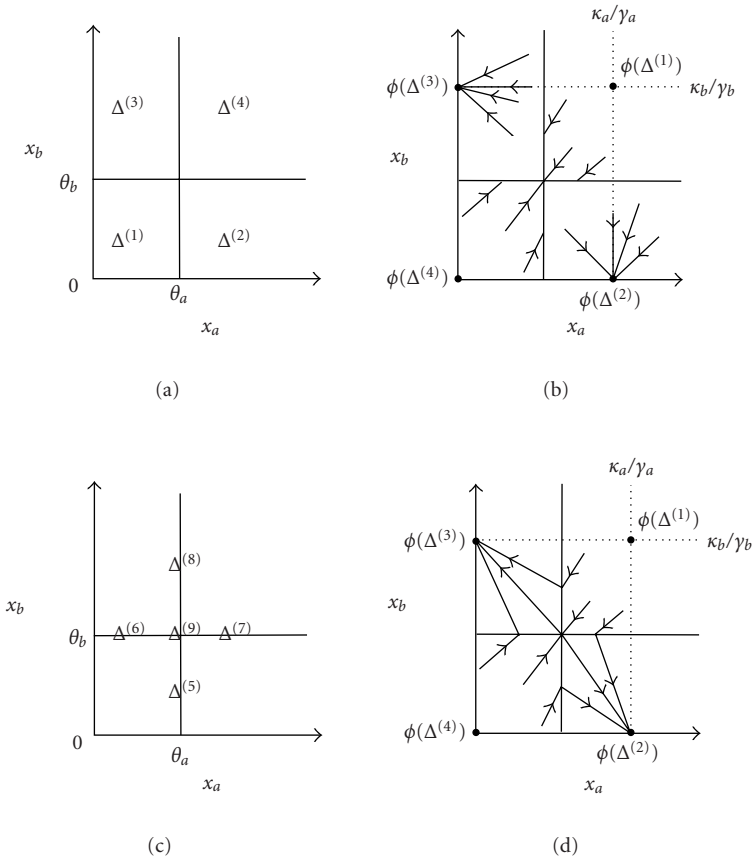


Figure 6.14. Local phase portraits of the piecewise-linear model of the mutual-inhibition network (Figure 6.13). (a) Regions $\Delta^{(1)},\ldots,\Delta^{(4)}$, (b) examples of trajectories in these regions, (c) regions $\Delta^{(5)},\ldots,\Delta^{(9)}$ located on the segments $x_a = \theta_a$ or $x_b = \theta_b$, and (d) examples of trajectories arriving at or departing from these regions. The trajectories are straight lines, because in the simulations we have set $\gamma_a = \gamma_b$.

In fact, every region of the phase space can be seen as a *qualitative state*, in which the system behaves in a qualitatively homogeneous way. For instance, in region $\Delta^{(1)}$, all trajectories converge towards the point $\phi(\Delta^{(1)}) = (\kappa_a/\gamma_a, \kappa_b/\gamma_b)'$, whereas in $\Delta^{(2)}$, they converge towards $\phi(\Delta^{(2)}) = (0, \kappa_b/\gamma_b)'$. Two qualitative states can be connected by a *transition*, if there exists a solution starting in the region corresponding to the first state that reaches the region corresponding to the second state, without passing through a third region. This is the case for the solutions in $\Delta^{(1)}$ which, while converging towards $\phi(\Delta^{(1)})$, reach $\Delta^{(5)}$, $\Delta^{(6)}$, or $\Delta^{(9)}$. The set of qualitative states and transitions between these states defines a *state transition graph*.

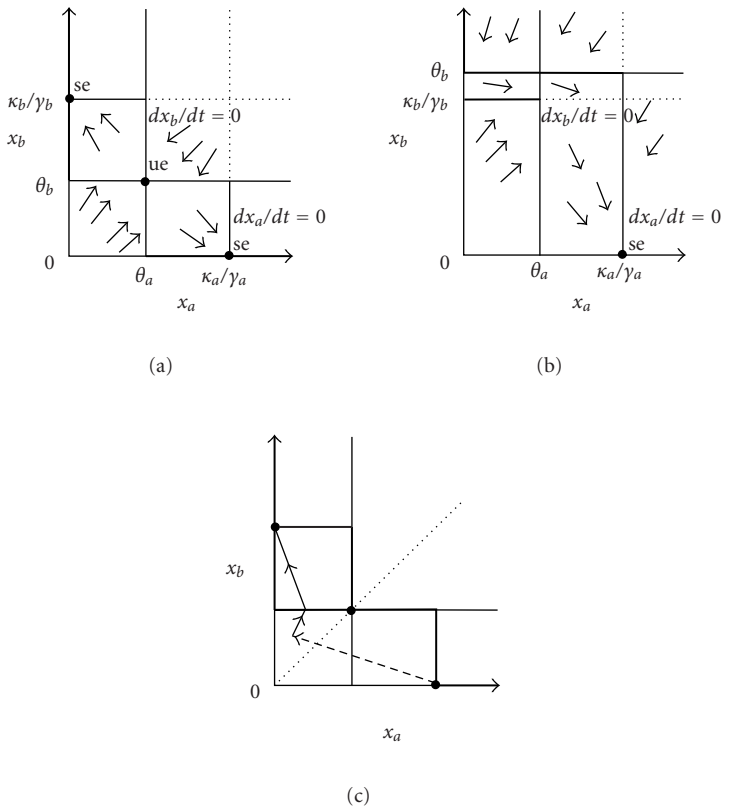


Figure 6.15. Global phase portrait of the piecewise-linear model of the mutual-inhibition network (Figure 6.13). (a) Vector field and nullclines. The system has two stable equilibrium points (se) and one unstable equilibrium point (ue). (b) Analysis of the bifurcation produced when the value of parameter θ_b is increased. The value of θ_b in (b) is larger than that in (a). (c) Hysteresis phenomenon, following a transient perturbation of the system (broken line with arrow).

The state transition graph obtained for the model of the mutual-inhibition network is shown in Figure 6.16a. The graph is composed of nine qualitative states, associated to the regions of the phase space (Figure 6.14), and the transitions between these states. Three of the nine states are *qualitative equilibrium states*, that is, states corresponding to a region containing an equilibrium point. The graph summarizes the dynamics of the network in a qualitative manner. For instance, it provides information on the reachability of an equilibrium point from a given region. If the equilibrium point is reachable, there must exist a path in the graph going from the qualitative state corresponding to the initial region to the qualitative equilibrium state corresponding to the region in which the equilibrium point is contained.

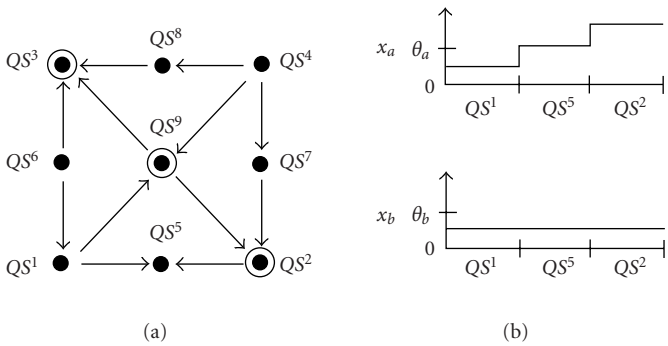


Figure 6.16. (a) State transition graph produced from the piecewise-linear model of the mutual-inhibition network (Figure 6.13). The qualitative equilibrium states are circled [69]. (b) Detailed description of the sequence of qualitative states QS^1, QS^5, QS^2.

Generally speaking, the state transition graph associated to a piecewise-linear model will vary with the parameter values. However, following de Jong et al. [69], one can define a class of models determined by inequality constraints on the parameters. Under certain, not too restrictive conditions, each model in that class produces the same state transition graph. This can be illustrated by means of the example of the two-gene network, by considering the transitions from QS^1, the qualitative state corresponding to region $\Delta^{(1)}$. The transitions from QS^1 to the states QS^5, QS^6, and QS^9 do not depend on the exact values of the parameters, as long as $\kappa_a/\gamma_a > \theta_a$ and $\kappa_b/\gamma_b > \theta_b$. In fact, under these conditions, $\phi(\Delta^{(1)}) \in \Delta^{(4)}$ and the trajectories in $\Delta^{(1)}$ all reach QS^5, QS^6, or QS^9 after a certain time. A *qualitative simulation* method has been proposed, which symbolically computes the state transition graph for a piecewise-linear differential equation model, supplemented by inequality constraints on the parameters [69]. This method has been implemented in a computer tool called Genetic Network Analyzer (GNA) [70].

The interest of qualitative simulation derives from the fact that it is adapted to the lack of quantitative information on genetic regulatory networks, a problem already referred to in previous sections. Instead of numerical values, the method uses inequality constraints that can usually be specified by means of the qualitative information available in the experimental literature. On the formal level, the qualitative simulation method is related to a method developed by Thomas et al., which is based on asynchronous logical models [13, 71]. A similar approach has also emerged in the hybrid-system community [72, 73]. The above methods have demonstrated their usefulness in the study of a certain number of prokaryotic and eukaryotic networks, whose analysis is rendered difficult by the almost complete absence of numerical parameter values (see [73, 74, 75, 76, 77, 78]; see [79] for a review). In the next section, we present the results of the qualitative simulation of the network controling the initiation of sporulation in *Bacillus subtilis*.

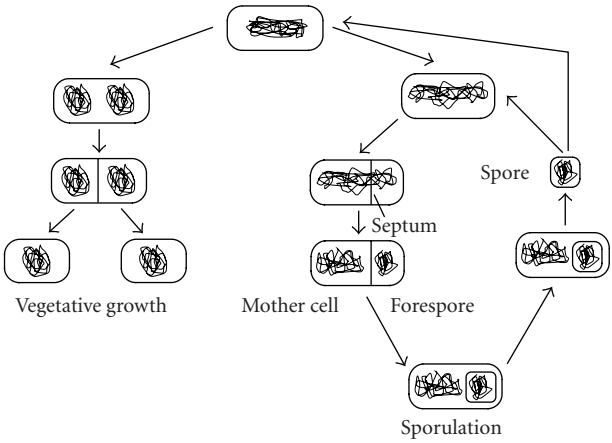


Figure 6.17. Life cycle of *B. subtilis*: decision between vegetative growth and sporulation (adapted from [81]).

6.5.2. Simulation of the initiation of sporulation in *B. subtilis*

Under conditions of nutrient deprivation, the Gram positive soil bacterium *B. subtilis* can abandon vegetative growth and form a dormant, environmentally resistant spore instead [80]. During vegetative growth, the cell divides symmetrically and generates two identical cells. During sporulation, on the other hand, cell division is asymmetric and results in two different cell types: the smaller cell (the forespore) develops into the spore, whereas the larger cell (the mother cell) helps to deposit a resistant coat around the spore and then disintegrates (Figure 6.17).

The decision to abandon vegetative growth and initiate sporulation involves a radical change in the genetic program, the pattern of gene expression, of the cell. The switch of genetic program is controled by a complex genetic regulatory network integrating various environmental, cell-cycle, and metabolic signals. Due to the ease of genetic manipulation of *B. subtilis*, it has been possible to identify and characterize a large number of the genes, proteins, and interactions making up this network. Currently, more than 125 genes are known to be involved [82]. A graphical representation of the regulatory network controlling the initiation of sporulation is shown in Figure 6.18, displaying key genes and their promoters, proteins encoded by the genes, and the regulatory action of the proteins.

The network is centered around a *phosphorelay*, which integrates a variety of environmental, cell-cycle, and metabolic signals. Under conditions appropriate for sporulation, the phosphorelay transfers a phosphate to the Spo0A regulator, a process modulated by kinases and phosphatases. The phosphorelay has been simplified in this paper by ignoring intermediate steps in the transfer of phosphate to Spo0A. However, this simplification does not affect the essential function of the phosphorelay: modulating the phosphate flux as a function of the competing action of kinases and phosphatases (here KinA and Spo0E). Under conditions conducive to sporulation, such as nutrient deprivation or high population density, the concentration of phosphorylated Spo0A (Spo0A~P) may reach a threshold value

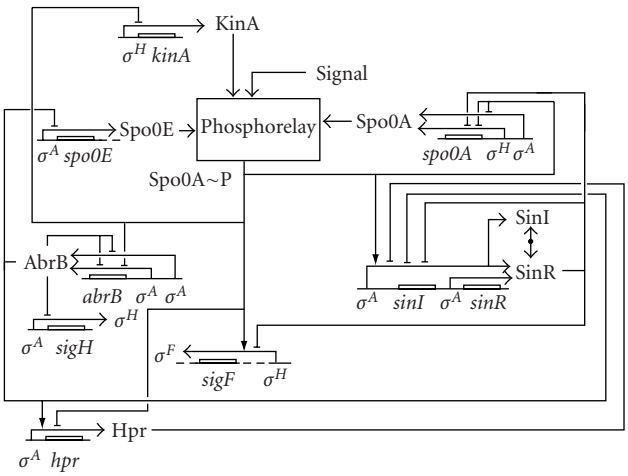


Figure 6.18. Key genes, proteins, and regulatory interactions making up the network involved in *B. subtilis* sporulation. In order to improve the legibility of the figure, the control of transcription by the sigma factors σ^A and σ^H has been represented implicitly, by annotating the promoter with the corresponding sigma factor (figure reproduced from [74]).

above which it activates various genes that commit the bacterium to sporulation. The choice between vegetative growth and sporulation in response to adverse environmental conditions is the outcome of competing positive and negative feedback loops, controling the accumulation of Spo0A~P.

Not withstanding the enormous amount of work devoted to the elucidation of the network of interactions underlying the sporulation process, very little quantitative data on kinetic parameters and molecular concentrations are available. de Jong et al. have therefore used the qualitative simulation method introduced in Section 6.5.1 to analyze the network [74]. The objective of the study was to reproduce the observed qualitative behavior of wild-type and mutant bacteria from a model synthesizing data available in the literature. To this end, the graphical representation of the network has been translated into a piecewise-linear model supplemented by qualitative constraints on the parameters. The resulting model consists of nine state variables and two input variables. The 48 parameters are constrained by 70 parameter inequalities, the choice of which is largely determined by biological data.

The tool GNA [70] has been used to simulate the response of a wild-type *B. subtilis* cell to nutrient depletion and high population density. Starting from initial conditions representing vegetative growth, the system is perturbed by a sporulation signal that causes KinA to autophosphorylate. Simulation of the network takes less than a few seconds to complete on a PC (500 MHz, 128 MB of RAM), and gives rise to a transition graph of 465 qualitative states. Many of these states are associated with regions in the phase space that the system traverses instantaneously. Since the biological relevance of the latter states is limited, they can be eliminated from the transition graph. This leads to a reduced transition graph with 82 qualitative states.

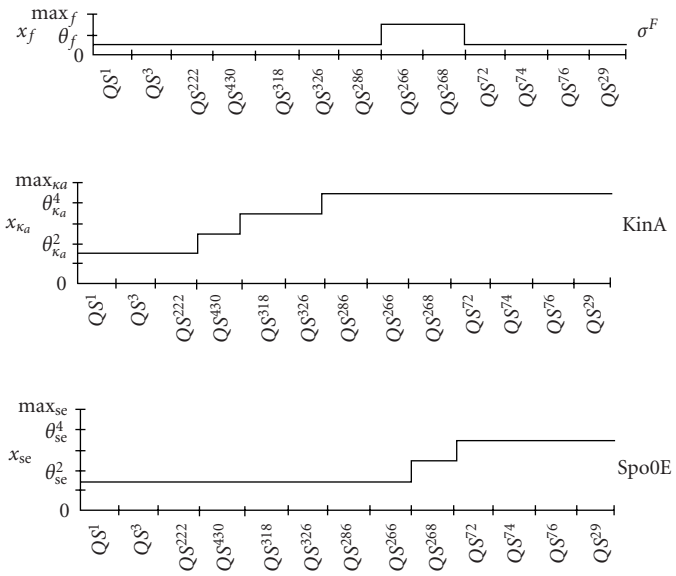
The transition graph faithfully represents two possible responses to nutrient depletion that are observed for *B. subtilis*: either the bacterium continues vegetative growth or it enters sporulation. Sequences of qualitative states typical for these two developmental modes are shown in Figure 6.19. The initiation of sporulation is determined by positive feedback loops acting through Spo0A and KinA, and a negative feedback loop involving Spo0E. If the rate of accumulation of the kinase KinA outpaces the rate of accumulation of the phosphatase Spo0E, we observe transient expression of *sigF*, that is, a spo^+ phenotype (Figure 6.19a). Gene *sigF* is a sigma factor essential for the development of the forespore [83]. If the kinetics of these processes are inversed, *sigF* is never activated and we observe a spo^- phenotype (Figure 6.19b). Deletion or overexpression of genes in the network of Figure 6.18 may disable a feedback loop, leading to specific changes in the observed sporulation phenotype. The results of the simulation of a dozen sporulation mutants are discussed in [74].

6.6. Discussion

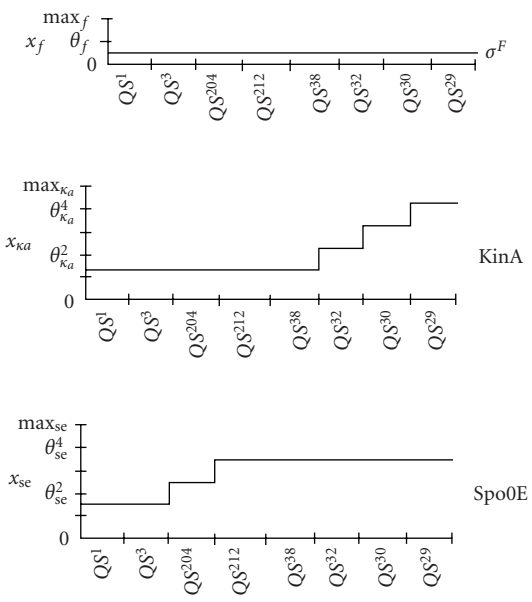
The functioning and development of living things—from bacteria to humans—are controled by genetic regulatory networks composed of interactions between DNA, RNA, proteins, and small molecules. The size and complexity of these networks make it difficult to intuitively understand their dynamics. In order to predict the behavior of regulatory systems in a systematic way, we need modeling and simulation tools with a solid foundation in mathematics, statistics, and computer science. In this chapter, we have examined the modeling and simulation of genetic regulatory networks by means of ordinary differential equations. In particular, we have compared nonlinear, linear, and piecewise-linear differential equation models. Concrete examples, taken from the literature, have illustrated the application of these models.

Nonlinear ordinary differential equations provide an adequate description of the dynamics of a genetic regulatory network, as shown by the analysis of the mutual-inhibition network. Unfortunately, they are difficult to treat mathematically for networks comprising more than two genes, in which case we have to take recourse to numerical simulation. However, the application of numerical techniques is often difficult in practice, due to the absence of numerical values for the parameters in the model. A possible alternative is the use of linear ordinary differential equations. Powerful techniques for solving these equations exist, as well as techniques for estimating parameter values from experimental data. This facility comes at a price though: the models are often too simple for reproducing essential dynamical properties of genetic regulatory networks.

In addition, we have presented a third type of ordinary differential equations, piecewise-linear models. Globally nonlinear and locally linear, these models represent a good compromise between the two types of models previously discussed. On one hand, they allow the network dynamics to be described in an adequate way, while on the other hand, they are easy to treat mathematically. The favorable mathematical properties have allowed the elaboration of a qualitative simulation



(a)



(b)

Figure 6.19. (a) Temporal evolution of selected protein concentrations in a typical qualitative behavior corresponding to the *spo*$^+$ phenotype. (b) Idem, but for a typical qualitative behavior corresponding to the *spo*$^-$ phenotype (figure adapted from [74]).

method, capable of producing a qualitative description of the dynamics of the network from a piecewise-linear differential equation model supplemented by inequality constraints on the parameters. Contrary to numerical values, these constraints can be often inferred from the experimental literature.

The ordinary differential equation models discussed in this chapter should not be mistaken for a panacea for the analysis of genetic regulatory networks. In fact, the applicability of these models can be questioned for different reasons and, based on these criticisms, other modeling approaches can be advanced. Here we concentrate on two objections.

First, the biochemical reactions occurring in a cell are discrete and stochastic processes [18]. The time between different reactions, as well as the type of the next reaction, is random, and each reaction increases and/or decreases the number of each molecular species. Ordinary differential equation models abstract from this molecular vision by reasoning in terms of continuous concentration variables and deterministic rates. This is an adequate approximation as long as the number of molecules is high. However, this condition is not always verified [18]. A bacterial cell, for instance, contains only a few dozens of molecules of certain transcription factors. In order to treat this problem, *stochastic master equation* models, based on the discrete and stochastic character of biochemical reaction systems, have been proposed (Section 6.1). The practical application of these models demands *stochastic simulation* methods [17, 84, 85], supported by efficient computer tools [86, 87]. The use of stochastic simulation has given rise to impressive results [16, 85], but has the disadvantage of demanding as input detailed information on the biochemical reactions as well as the associated kinetic parameters, information that is only seldom available. Moreover, its application to regulatory networks of more than a few genes requires huge computational resources.

The second objection is not directed at the validity of the models encountered in this chapter, but rather concerns the feasibility of their application. It takes as its point of departure the observation that the genetic regulatory networks of interest generally involve a large number of genes and proteins, interacting in complex ways. If one wishes to model these networks in the same way as the example network of two genes, the resulting models will have hundreds of variables and parameters. Given the difficulties of obtaining reliable parameter values, the conclusions that one will be able to draw from the analysis of these gigantic models have to be taken with circumspection. Moreover, it is not sure that the predictions of the temporal evolution of all these concentration variables will help us to better understand the functioning of the system. In this case, it seems more appropriate and informative to distinguish subnetworks of the network of interest, describe the dynamics of these subnetworks individually—using more abstract models than those presented here—and couple the abstract models in order to analyze the interactions between subnetworks [88, 89]. This *modular* approach can be justified by what we know or suspect about the structure of biological regulatory networks [55, 90].

The two objections seem to point in two opposite directions: towards more fine-grained models for the first objection, and towards more abstract models for

the second objection. However, the contradiction is only apparent; it disappears when one recalls that a model is constructed in order to answer a certain biological question. If one is interested in the analysis of the behavior emerging from the interactions between several dozens of genes, the ordinary differential equation models presented in this chapter are quite appropriate. On the contrary, a more fine-grained analysis of a particular regulatory mechanism may demand the use of stochastic models, while a global comprehension of a very large network is probably better approached by means of approximate models, describing the interactions between different modules of the system.

This perspective emphasizes the importance of the *modeling* task, the construction of a model adapted to the question being asked, as well as the revision of the model when one discovers that it is not adequate (Figure 6.1). Globally, there exist two approaches for model construction [6]. On one hand, they can be composed from knowledge on the molecular components of the regulatory system and their mutual interactions, whereas on the other hand, they can be inferred from expression data and other measurements of the kinetics of the system. In practice, the two approaches must be combined in order to be able to efficiently exploit the rich store of information available on the structure and functioning of genetic regulatory networks. One of the big challenges of bioinformatics, statistics, and system biology today consists in the development of computer environments capable of supporting the construction of simulation models [91].

Another challenge consists in the integration of genetic regulatory networks with metabolic networks, signal transduction networks, and other interaction networks (Section 6.1). Even if, for certain problems, the study of one type of networks in isolation may be satisfactory, the comprehension of the functioning of an entire cell obliges us to build models combining gene regulation with metabolism, signal transduction, and other processes. Excellent mathematical models of different cellular processes exist nowadays, for example of the cell cycle in the toad *Xenopus laevis* and the yeasts *Schizosaccharomyces pombe* and *Saccharomyces cerevisiae* [92, 93, 94, 95], the metabolism of the red-blood cell in humans [96, 97], and the signaling pathway involved in the maturation of oocytes in *X. laevis* [98, 99]. However, the integration of models of different processes of the cell remains a very difficult task. As remarked in the introductory section, the networks involve different types of interaction, modeled by different types of equations. Moreover, the processes that are concerned evolve on different time scales, sometimes differing by several orders of magnitude. Among other things, this raises mathematical problems associated with the stiffness of the resulting differential equations. Several approaches for the integration of different types of networks have been proposed (e.g., [100]), but there can be no doubt that the subject remains largely unexplored.

Acknowledgment

The authors would like to thank Jean-Luc Gouzé, Michel Page, and Delphine Ropers for their comments on a previous version of this chapter.

Bibliography

[1] D. J. Lockhart and E. A. Winzeler, "Genomics, gene expression and DNA arrays," *Nature*, vol. 405, no. 6788, pp. 827–836, 2000.

[2] A. Pandey and M. Mann, "Proteomics to study genes and genomes," *Nature*, vol. 405, no. 6788, pp. 837–846, 2000.

[3] T. Ideker, T. Galitski, and L. Hood, "A new approach to decoding life: systems biology," *Annu. Rev. Genomics Hum. Genet.*, vol. 2, pp. 343–372, 2001.

[4] H. Kitano, "Systems biology: a brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.

[5] J. M. Bower and H. Bolouri, Eds., *Computational Modeling of Genetic and Biochemical Networks*, MIT Press, Cambridge, Mass, USA, 2001.

[6] H. de Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *J. Comput. Biol.*, vol. 9, no. 1, pp. 67–103, 2002.

[7] A. Gilman and A. P. Arkin, "Genetic "code": representations and dynamical models of genetic components and networks," *Annu. Rev. Genomics Hum. Genet.*, vol. 3, pp. 341–369, 2002.

[8] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: *in numero* molecular biology," *Nat. Rev. Genet.*, vol. 2, no. 4, pp. 268–279, 2001.

[9] P. Smolen, D. A. Baxter, and J. H. Byrne, "Modeling transcriptional control in gene networks—methods, recent results, and future directions," *Bull. Math. Biol.*, vol. 62, no. 2, pp. 247–292, 2000.

[10] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 601–620, 2000.

[11] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, NY, USA, 1993.

[12] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[13] R. Thomas and R. d'Ari, *Biological Feedback*, CRC Press, Boca Raton, Fla, USA, 1990.

[14] A. Cornish-Bowden, *Fundamentals of Enzyme Kinetics*, Portland Press, London, UK, revised edition, 1995.

[15] R. Heinrich and S. Schuster, *The Regulation of Cellular Systems*, Chapman & Hall, New York, NY, USA, 1996.

[16] A. Arkin, J. Ross, and H. H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells," *Genetics*, vol. 149, no. 4, pp. 1633–1648, 1998.

[17] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[18] H. H. McAdams and A. Arkin, "It's a noisy business! Genetic regulation at the nanomolar scale," *Trends Genet.*, vol. 15, no. 2, pp. 65–69, 1999.

[19] T. M. Gruber and C. A. Gross, "Multiple sigma subunits and the partitioning of bacterial transcription space," *Annu. Rev. Microbiol.*, vol. 57, pp. 441–466, 2003.

[20] A. Ishihama, "Functional modulation of *Escherichia coli* RNA polymerase," *Annu. Rev. Microbiol.*, vol. 54, pp. 499–518, 2000.

[21] R. Hengge-Aronis, "Signal transduction and regulatory mechanisms involved in control of the σ^S (RpoS) subunit of RNA polymerase," *Microbiol. Mol. Biol. Rev.*, vol. 66, no. 3, pp. 373–395, 2002.

[22] H. Tagami and H. Aiba, "Role of CRP in transcription activation at *Escherichia coli lac* promoter: CRP is dispensable after the formation of open complex," *Nucleic Acids Res.*, vol. 23, no. 4, pp. 599–605, 1995.

[23] M. Cashel, D. R. Gentry, V. J. Hernandez, and D. Vinella, "The stringent response," in *Escherichia coli and Salmonella: Cellular and Molecular Biology*, F. C. Neidhardt, R. Curtiss III, J. L. Ingraham, et al., Eds., pp. 1458–1496, ASM Press, Washington, DC, USA, 2nd edition, 1996.

[24] M. Hirsch and T. Elliott, "Role of ppGpp in *rpoS* stationary-phase regulation in *Escherichia coli*," *J. Bacteriol.*, vol. 184, no. 18, pp. 5077–5087, 2002.

[25] R. Lange, D. Fischer, and R. Hengge-Aronis, "Identification of transcriptional start sites and the role of ppGpp in the expression of *rpoS*, the structural gene for the σ^S subunit of RNA polymerase in *Escherichia coli*," *J. Bacteriol.*, vol. 177, no. 16, pp. 4676–4680, 1995.

[26] V. Venturi, "Control of *rpoS* transcription in *Escherichia coli* and *Pseudomonas*: why so different?," *Mol. Microbiol.*, vol. 49, no. 1, pp. 1–9, 2003.

[27] J. P. Zhang and S. Normark, "Induction of gene expression in *Escherichia coli* after pilus-mediated adherence," *Science*, vol. 273, no. 5279, pp. 1234–1236, 1996.

[28] K. W. Kohn, "Molecular interaction maps as information organizers and simulation guides," *Chaos*, vol. 11, no. 1, pp. 84–97, 2001.

[29] F. Repoila, N. Majdalani, and S. Gottesman, "Small non-coding RNAs, co-ordinators of adaptation processes in *Escherichia coli*: the RpoS paradigm," *Mol. Microbiol.*, vol. 48, no. 4, pp. 855–861, 2003.

[30] A. Balandina, L. Claret, R. Hengge-Aronis, and J. Rouviere-Yaniv, "The *Escherichia coli* histone-like protein HU regulates rpoS translation," *Mol. Microbiol.*, vol. 39, no. 4, pp. 1069–1079, 2001.

[31] G. Becker, E. Klauck, and R. Hengge-Aronis, "Regulation of RpoS proteolysis in *Escherichia coli*: the response regulator RssB is a recognition factor that interacts with the turnover element in RpoS," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 11, pp. 6439–6444, 1999.

[32] N. Ruiz, C. N. Peterson, and T. J. Silhavy, "RpoS-dependent transcriptional control of *sprE*: regulatory feedback loop," *J. Bacteriol.*, vol. 183, no. 20, pp. 5974–5981, 2001.

[33] L. Zhou, X. H. Lei, B. R. Bochner, and B. L. Wanner, "Phenotype microarray analysis of *Escherichia coli* K-12 mutants with deletions of all two-component systems," *J. Bacteriol.*, vol. 185, no. 16, pp. 4956–4972, 2003.

[34] M. Jishage, K. Kvint, V. Shingler, and T. Nystrom, "Regulation of σ factor competition by the alarmone ppGpp," *Genes Dev.*, vol. 16, no. 10, pp. 1260–1270, 2002.

[35] H. E. Schellhorn and V. L. Stones, "Regulation of *katF* and *katE* in *Escherichia coli* K-12 by weak acids," *J. Bacteriol.*, vol. 174, no. 14, pp. 4769–4776, 1992.

[36] M. Ševčík, A. Šebková, J. Volf, and I. Rychlík, "Transcription of *arcA* and *rpoS* during growth of *Salmonella typhimurium* under aerobic and microaerobic conditions," *Microbiology*, vol. 147, no. Pt 3, pp. 701–708, 2001.

[37] P. Brazhnik, A. de la Fuente, and P. Mendes, "Gene networks: how to put the function in genomics," *Trends Biotechnol.*, vol. 20, no. 11, pp. 467–472, 2002.

[38] J. Monod and F. Jacob, "General conclusions: Teleonomic mechanisms in cellular metabolism, growth, and differentiation," in *Cold Spring Harbor Symposium on Quantitative Biology*, pp. 389–401, Cold Spring Harbor, NY, USA, 1961.

[39] M. Ptashne, *A Genetic Switch: Phage λ and Higher Organisms*, Cell Press & Blackwell Publishers, Cambridge, Mass, USA, 2nd edition, 1992.

[40] G. Yagil and E. Yagil, "On the relation between effector concentration and the rate of induced enzyme synthesis," *Biophys. J.*, vol. 11, no. 1, pp. 11–27, 1971.

[41] D. Kaplan and L. Glass, *Understanding Nonlinear Dynamics*, vol. 19 of *Texts in Applied Mathematics*, Springer-Verlag, New York, NY, USA, 1995.

[42] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in *Escherichia coli*," *Nature*, vol. 403, no. 6767, pp. 339–342, 2000.

[43] J. D. Lambert, *Numerical Methods for Ordinary Differential Equations*, Wiley, Chichester, UK, 1991.

[44] P. Mendes, "GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems," *Comput. Appl. Biosci.*, vol. 9, no. 5, pp. 563–571, 1993.

[45] I. Goryanin, T. C. Hodgman, and E. Selkov, "Mathematical simulation and analysis of cellular metabolism and regulation," *Bioinformatics*, vol. 15, no. 9, pp. 749–758, 1999.

[46] E. Meir, E. M. Munro, G. M. Odell, and G. von Dassow, "Ingeneue: a versatile tool for reconstituting genetic networks, with examples from the segment polarity network," *J. Exp. Zool.*, vol. 294, no. 3, pp. 216–251, 2002.

[47] M. Hucka, A. Finney, H. M. Sauro, et al., "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, no. 4, pp. 524–531, 2003.

[48] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. Doyle, and H. Kitano, "The ERATO systems biology workbench: Enabling interactions and exchange between software tools for computational biology," in *Pacific Symposium on Biocomputing*, R. B. Altman, A. K. Dunker, L. Hunter, K. Lauderdale, and T. E. Klein, Eds., pp. 450–461, World Scientific Publishing, Singapore, 2002.

[49] L. Ljung, *System Identification: Theory for the User*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.

[50] P. A. Lawrence, *The Making of a Fly: The Genetics of Animal Design*, Blackwell Publishers, Oxford, UK, 1992.

[51] B. Sanson, "Generating patterns from fields of cells. Examples from Drosophila segmentation," *EMBO Rep.*, vol. 2, no. 12, pp. 1083–1088, 2001.

[52] G. von Dassow, E. Meir, E. M. Munro, and G. M. Odell, "The segment polarity network is a robust developmental module," *Nature*, vol. 406, no. 6792, pp. 188–192, 2000.

[53] E. Meir, G. von Dassow, E. Munro, and G. M. Odell, "Robustness, flexibility, and the role of lateral inhibition in the neurogenic network," *Curr. Biol.*, vol. 12, no. 10, pp. 778–786, 2002.

[54] N. Barkai and S. Leibler, "Robustness in simple biochemical networks," *Nature*, vol. 387, no. 6636, pp. 913–917, 1997.

[55] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, no. 6761 Suppl, pp. C47–C52, 1999.

[56] W. J. Rugh, *Linear System Theory*, Prentice Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1996.

[57] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[58] G. C. Walker, "The SOS system of *Escherichia coli*," in *Escherichia coli and Salmonella: Cellular and Molecular Biology*, F. C. Neidhardt, R. Curtiss III, J. L. Ingraham, et al., Eds., pp. 1400–1416, ASM Press, Washington, DC, USA, 2nd edition, 1996.

[59] G. C. Walker, B. T. Smith, and M. D. Sutton, "The SOS response to DNA damage," in *Bacterial Stress Responses*, G. Storz and R. Hengge-Aronis, Eds., pp. 131–144, ASM Press, Washington, DC, USA, 2000.

[60] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp. 102–105, 2003.

[61] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 16, pp. 10555–10560, 2002.

[62] A. K. Pernestig, D. Georgellis, T. Romeo, et al., "The *Escherichia coli* BarA-UvrY two-component system is needed for efficient switching between glycolytic and gluconeogenic carbon sources," *J. Bacteriol.*, vol. 185, no. 3, pp. 843–853, 2003.

[63] A. K. Pernestig, , O. Melefors, and D. Georgellis, "Identification of UvrY as the cognate response regulator for the BarA sensor kinase in *Escherichia coli*," *J. Biol. Chem.*, vol. 276, no. 1, pp. 225–231, 2001.

[64] R. Edwards, H. T. Siegelmann, K. Aziza, and L. Glass, "Symbolic dynamics and computation in model gene networks," *Chaos*, vol. 11, no. 1, pp. 160–169, 2001.

[65] L. Glass and S. A. Kauffman, "The logical analysis of continuous, non-linear biochemical control networks," *J. Theor. Biol.*, vol. 39, no. 1, pp. 103–129, 1973.

[66] T. Mestl, E. Plahte, and S. W. Omholt, "A mathematical framework for describing and analysing gene regulatory networks," *J. Theor. Biol.*, vol. 176, no. 2, pp. 291–300, 1995.

[67] E. H. Snoussi, "Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach," *Dynam. Stability Systems*, vol. 4, no. 3-4, pp. 189–207, 1989.

[68] J.-L. Gouzé and T. Sari, "A class of piecewise linear differential equations arising in biological models," *Dyn. Syst.*, vol. 17, no. 4, pp. 299–316, 2002.

[69] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann, "Qualitative simulation of genetic regulatory networks using piecewise-linear models," *Bull. Math. Biol.*, vol. 66, no. 2, pp. 301–340, 2004.

[70] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page, "Genetic Network Analyzer: qualitative simulation of genetic regulatory networks," *Bioinformatics*, vol. 19, no. 3, pp. 336–344, 2003.

[71] R. Thomas, D. Thieffry, and M. Kaufman, "Dynamical behaviour of biological regulatory networks–I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state," *Bull. Math. Biol.*, vol. 57, no. 2, pp. 247–276, 1995.

[72] R. Ghosh, A. Tiwari, and C. L. Tomlin, "Automated symbolic reachability analysis with application to Delta-Notch signaling automata," in *Hybrid Systems: Computation and Control*, O. Maler and A. Pnueli, Eds., vol. 2623 of *Lecture Notes in Computer Science*, pp. 233–248, Springer-Verlag, Berlin, Germany, 2003.

[73] R. Ghosh and C. J. Tomlin, "Lateral inhibition through Delta-Notch signaling: A piecewise affine hybrid model," in *Hybrid Systems: Computation and Control*, M. D. Di Benedetto and A. Sangiovanni-Vincentelli, Eds., vol. 2034 of *Lecture Notes in Computer Science*, pp. 232–246, Springer-Verlag, Berlin, Germany, 2001.

[74] H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page, "Qualitative simulation of the initiation of sporulation in *Bacillus subtilis*," *Bull. Math. Biol.*, vol. 66, no. 2, pp. 261–299, 2004.

[75] L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla, "Genetic control of flower morphogenesis in *Arabidopsis thaliana*: a logical analysis," *Bioinformatics*, vol. 15, no. 7-8, pp. 593–606, 1999.

[76] L. Sánchez and D. Thieffry, "A logical analysis of the *Drosophila* gap-gene system," *J. Theor. Biol.*, vol. 211, no. 2, pp. 115–141, 2001.

[77] L. Sánchez and D. Thieffry, "Segmenting the fly embryo: a logical analysis of the pair-rule cross-regulatory module," *J. Theor. Biol.*, vol. 224, no. 4, pp. 517–537, 2003.

[78] D. Thieffry and R. Thomas, "Dynamical behaviour of biological regulatory networks–II. Immunity control in bacteriophage lambda," *Bull. Math. Biol.*, vol. 57, no. 2, pp. 277–297, 1995.

[79] H. de Jong, J. Geiselmann, and D. Thieffry, "Qualitative modelling and simulation of developmental regulatory networks," in *On Growth, Form, and Computers*, S. Kumar and P. J. Bentley, Eds., pp. 109–134, Academic Press, London, UK, 2003.

[80] W. F. Burkholder and A. D. Grossman, "Regulation of the initiation of endospore formation in *Bacillus subtilis*," in *Prokaryotic Development*, Y. V. Brun and L. J. Shimkets, Eds., chapter 7, pp. 151–166, American Society for Microbiology, Washington, DC, USA, 2000.

[81] P. A. Levin and A. D. Grossman, "Cell cycle and sporulation in *Bacillus subtilis*," *Curr. Opin. Microbiol.*, vol. 1, no. 6, pp. 630–635, 1998.

[82] P. Fawcett, P. Eichenberger, R. Losick, and P. Youngman, "The transcriptional profile of early to middle sporulation in *Bacillus subtilis*," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 14, pp. 8063–8068, 2000.

[83] P. Stragier and R. Losick, "Molecular genetics of sporulation in *Bacillus subtilis*," *Annu. Rev. Genet.*, vol. 30, pp. 297–341, 1996.

[84] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *Journal of Physical Chemistry A*, vol. 104, no. 9, pp. 1876–1889, 2000.

[85] C. J. Morton-Firth and D. Bray, "Predicting temporal fluctuations in an intracellular signalling pathway," *J. Theor. Biol.*, vol. 192, no. 1, pp. 117–128, 1998.

[86] A. M. Kierzek, "STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm," *Bioinformatics*, vol. 18, no. 3, pp. 470–481, 2002.

[87] N. Le Novère and T. S. Shimizu, "STOCHSIM: modelling of stochastic biomolecular processes," *Bioinformatics*, vol. 17, no. 6, pp. 575–576, 2001.

[88] U. S. Bhalla and R. Iyengar, "Emergent properties of networks of biological signaling pathways," *Science*, vol. 283, no. 5400, pp. 381–387, 1999.

[89] B. N. Kholodenko, A. Kiyatkin, F. J. Bruggeman, E. Sontag, H. V. Westerhoff, and J. B. Hoek, "Untangling the wires: a strategy to trace functional interactions in signaling and gene networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 20, pp. 12841–12846, 2002.

[90] D. Thieffry, A. M. Huerta, E. Pérez-Rueda, and J. Collado-Vides, "From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*," *BioEssays*, vol. 20, no. 5, pp. 433–440, 1998.

[91] H. de Jong and A. Rip, "The computer revolution in science: steps towards the realization of computer-supported discovery environments," *Artificial Intelligence*, vol. 91, no. 2, pp. 225–256, 1997.

[92] K. C. Chen, A. Csikasz-Nagy, B. Gyorffy, J. Val, B. Novak, and J. J. Tyson, "Kinetic analysis of a molecular model of the budding yeast cell cycle," *Mol. Biol. Cell*, vol. 11, no. 1, pp. 369–391, 2000.
[93] B. Novak and J. J. Tyson, "Numerical analysis of a comprehensive model of M-phase control in Xenopus oocyte extracts and intact embryos," *J. Cell Sci.*, vol. 106, no. Pt 4, pp. 1153–1168, 1993.
[94] J. J. Tyson, "Models of cell cycle control in eukaryotes," *J. Biotechnol.*, vol. 71, no. 1-3, pp. 239–244, 1999.
[95] J. J. Tyson, K. Chen, and B. Novak, "Network dynamics and cell physiology," *Nat. Rev. Mol. Cell Biol.*, vol. 2, no. 12, pp. 908–916, 2001.
[96] N. Jamshidi, J. S. Edwards, T. Fahland, G. M. Church, and B. O. Palsson, "Dynamic simulation of the human red blood cell metabolic network," *Bioinformatics*, vol. 17, no. 3, pp. 286–287, 2001.
[97] A. Joshi and B. O. Palsson, "Metabolic dynamics in the human red cell. Part I–a comprehensive kinetic model," *J. Theor. Biol.*, vol. 141, no. 4, pp. 515–528, 1989.
[98] J. E. Ferrell Jr. and E. M. Machleder, "The biochemical basis of an all-or-none cell fate switch in *Xenopus* oocytes," *Science*, vol. 280, no. 5365, pp. 895–898, 1998.
[99] J. E. Ferrell and W. Xiong, "Bistability in cell signaling: How to make continuous processes discontinuous, and reversible processes irreversible," *Chaos*, vol. 11, no. 1, pp. 227–236, 2001.
[100] M. W. Covert, C. H. Schilling, and B. Palsson, "Regulation of gene expression in flux balance models of metabolism," *J. Theor. Biol.*, vol. 213, no. 1, pp. 73–88, 2001.

Hidde de Jong: Institut National de Recherche en Informatique et en Automatique (INRIA), Unité de Recherche Rhône-Alpes, 655 avenue de l'Europe, Montbonnot, 38334 Saint Ismier Cedex, France
Email: hidde.de-jong@inrialpes.fr

Johannes Geiselmann: Laboratoire Adaptation et Pathogénie des Microorganismes Université Joseph Fourier (CNRS UMR 5163), Bâtiment Jean Roget, Domaine de la Merci, 38700 la Tronche, France
Email: hans.geiselmann@ujf-grenoble.fr

7 Modeling genetic regulatory networks with probabilistic Boolean networks

Ilya Shmulevich and Edward R. Dougherty

7.1. Introduction

High-throughput genomic technologies such as microarrays are now allowing scientists to acquire extensive information on gene activities of thousands of genes in cells at any physiological state. It has long been known that genes and their products in cells are not independent in the sense that the activation of genes with subsequent production of proteins is typically jointly dependent on the products of other genes, which exist in a highly interactive and dynamic regulatory network composed of subnetworks and regulated by rules. However, discovering the network structure has thus far proved to be elusive either because we lack sufficient information on the components of the network or because we lack the necessary multidisciplinary approaches that integrate biology and engineering principles and computational sophistication in modeling. During the past several years a new mathematical rule-based model called probabilistic Boolean networks (PBN) has been developed to facilitate the construction of gene regulatory networks to assist scientists in revealing the intrinsic gene-gene relationships in cells and in exploring potential network-based strategies for therapeutic intervention (Shmulevich et al. [1, 2, 3, 4, 5, 6], Datta et al. [7, 8], Kim et al. [9], Zhou et al. [10], and Hashimoto et al. [11]). There is already evidence that PBN models can reveal biologically relevant gene regulatory networks and can be used to predict the effects of targeted gene intervention. A key goal of this chapter is to highlight some important research problems related to PBNs that remain to be solved, in hope that they will stimulate further research in the genomic signal processing and statistics community.

7.2. Background

Data comprised of gene expression (mRNA abundance) levels for multiple genes is typically generated by technologies such as the DNA microarray or chip. The role

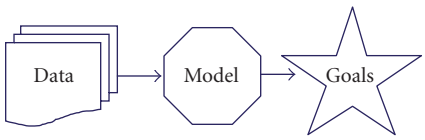


Figure 7.1

of the dynamical model or network is to simulate, via iteration of explicit rules, the dynamics of the underlying system presumed to be generating the observations. Such simulations can be useful for making predictions while the rules themselves, characterizing the relationships between the expressions of different genes, may hold important biological information. Time-course data, which are measurements taken at a number of time points, are often used for the inference of the model.

Before we discuss Probabilistic Boolean Networks, it may be worthwhile to pose several general but fundamental questions concerning modeling of genetic regulatory networks. The first and perhaps the most important question is the following.

7.2.1. What class of models should we choose?

We would like to argue that this choice must be made in view of (1) the data requirements and (2) the goals of modeling and analysis. Indeed, as shown in Figure 7.1 data is required to infer the model parameters from observations in the physical world, while the model itself must serve some purpose, in particular, prediction of certain aspects of the system under study. Simply put, the type and quantity of data that we can gather together with our prescribed purpose for using a model should be the main determining factors behind choosing a model class.

The choice of a model class involves a classical tradeoff. A *fine-scale* model with many parameters may be able to capture detailed *low-level* phenomena such as protein concentrations and reaction kinetics, but will require very large amounts of highly accurate data for its inference, in order to avert overfitting the model. In contrast, a *coarse-scale* model with lower complexity may succeed in capturing *high-level* phenomena, such as which genes are *on* or *off*, while requiring smaller amounts of more coarse-scale data. In the context of genetic regulatory systems, fine-scale models, typically involving systems of differential equations, can be applied to relatively small and isolated genetic circuits for which many types of accurate measurements can be made. On the other hand, coarse-scale models are more suited to global (genome-wide) measurements, such as those produced by microarrays. Such considerations should drive the selection of the model class. Needless to say, according to the principle of Ockham's razor, which underlies all scientific theory building, the model complexity should never be made higher than what is necessary to faithfully "explain the data" (Shmulevich [12]).

There is a rather wide spectrum of approaches for modeling gene regulatory networks, each with its own assumptions, data requirements, and goals, including linear models (van Someren et al. [13], D'haeseleer [14]), Bayesian networks (Murphy and Mian [15], Friedman et al. [16], Hartemink et al. [17], Moler et al. [18]), neural networks (Weaver et al. [19]), differential equations (Mestl et al. [20], Chen et al. [21], Goutsias and Kim [22]), as well as models including stochastic components on the molecular level (McAdams and Arkin [23]; Arkin et al. [24]) (see Smolen et al. [25], Hasty et al. [26], and de Jong [27] for reviews of general models).

The model system that has received, perhaps, the most attention is the Boolean network model originally introduced by Kauffman (Kauffman [28], Glass and Kauffman [29]). Good reviews can be found in Huang [30], Kauffman [31], Somogyi and Sniegoski [32], Aldana et al. [33]. In this model, the state of a gene is represented by a Boolean variable (on or off) and interactions between the genes are represented by Boolean functions, which determine the state of a gene on the basis of the states of some other genes.

One of the appealing properties of Boolean networks is that they are inherently simple, emphasizing generic principles rather than quantitative biochemical details, but are able to capture the complex dynamics of gene regulatory networks. Computational models that reveal these logical interrelations have been successfully constructed (Bodnar [34], Yuh et al. [35], Mendoza et al. [36], Huang and Ingber [37]). Let us now pose several questions related to this class of models.

7.2.2. To what extent do such models represent reality?

This question pertains more to modeling in general. All models only approximate reality by means of some formal representation. It is the degree to which we hope to approximate reality and, more importantly, our goals of modeling, namely, to acquire knowledge about some physical phenomenon that determines what class of models should be chosen. In the context of Boolean networks as models of genetic regulatory networks, the binary approximation of gene expression is only suitable to capture those aspects of regulation that possess a somewhat binary character. Even though most biological phenomena manifest themselves in the continuous domain, we often describe them in a binary logical language such as "on and off," "up-regulated and down-regulated," and "responsive and nonresponsive." Moreover, recent results suggest that gene regulation may indeed function "digitally" (Lahav et al. [38]). Before embarking on modeling gene regulatory networks with a Boolean formalism, it is prudent to test whether or not meaningful biological information can be extracted from gene expression data entirely in the binary domain. This question was taken up by Shmulevich and Zhang [39]. They reasoned that if the gene expression levels, when quantized to only two levels (1 or 0), would not be informative in separating known subclasses of tumors, then there would be little hope for Boolean modeling of realistic genetic networks based on gene expression data. Fortunately, the results

were very promising. By using binary gene expression data, generated via cDNA microarrays, and the Hamming distance as a similarity metric, they were able to show a clear separation between different subtypes of gliomas (a similar experiment was also performed for sarcomas), using multidimensional scaling. This seems to suggest that a good deal of meaningful biological information, to the extent that it is contained in the measured continuous-domain gene expression data, is retained when it is binarized. Zhou et al. [40] took a similar approach, but in the context of classification. The revealing aspect of their approach is that classification using binarized expressions proved to be only negligibly inferior to that using the original continuous expression values, the difference being that the genes derived via feature selection in the binary setting were different than the ones selected in the continuous. The expression values of those possessing binary-like behavior fell into bimodal distributions and these were naturally selected for classification.

7.2.3. Do we have the "right" type of data to infer these models?

With cDNA microarray data, it is widely recognized that reproducibility of measurements and between-slide variation is a major issue (Zhang et al. [41], Chen et al. [42], Kerr et al. [43]). Furthermore, genetic regulation exhibits considerable uncertainty on the biological level. Indeed, evidence suggests that this type of "noise" is in fact advantageous in some regulatory mechanisms (McAdams and Arkin [44]). Thus, from a practical standpoint, limited amounts of data and the noisy nature of the measurements can make useful quantitative inferences problematic, and a coarse-scale qualitative modeling approach seems to be justified. To put it another way, if our goals of modeling were to capture the genetic interactions with fine-scale quantitative biochemical details in a global large-scale fashion, then the data produced by currently available high-throughput genomic technologies would not be adequate for this purpose.

Besides the noise and lack of fine-scale data, another important concern is the design of dynamic networks using nondynamic data. If time-course data is available, then it is usually limited and the relation between the biological time-scale under which it has been observed and the transition routine of an inferred network is unknown. Moreover, most often the data being used to infer networks does not consist of time-course observations. In this situation, the usual assumption is that the data comes from the steady state of the system. There are inherent limitations to the design of dynamical systems from steady-state data. Steady-state behavior constrains the dynamical behavior, but does not determine it. Thus, while we might obtain good inference regarding the attractors, we may obtain poor inference relative to the steady-state distribution (see Section 7.4.2 for the definition of an attractor). Building a dynamical model from steady-state data is a kind of overfitting. It is for this reason that we view a designed network as providing a regulatory structure consistent with the observed steady-state behavior. If our main interest is in steady-state behavior, then it is reasonable to try to understand dynamical regulation corresponding to steady-state behavior.

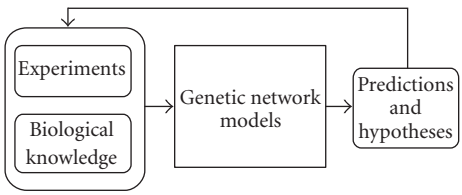


Figure 7.2

7.2.4. What do we hope to learn from these models?

Our last question is concerned with what type of knowledge we hope to acquire with the chosen models and the available data. As a first step, we may be interested in discovering qualitative relationships underlying genetic regulation and control. That is, we wish to emphasize fundamental generic coarse-grained properties of large networks rather than quantitative details, such as kinetic parameters of individual reactions (Huang [30]). Furthermore, we may wish to gain insight into the dynamical behavior of such networks and how it relates to underlying biological phenomena, such as cellular state dynamics, thus providing the potential for the discovery of novel targets for drugs. As an example, we may wish to predict the downstream effects of a targeted perturbation of a particular gene. Recent research indicates that many realistic biological questions may be answered within the seemingly simplistic Boolean formalism. Boolean networks are structurally simple, yet dynamically complex. They have yielded insights into the overall behavior of large genetic networks (Somogyi and Sniegoski [32], Szallasi and Liang [45], Wuensche [46], Thomas et al. [47]) and allowed the study of large data sets in a global fashion.

Besides the conceptual framework afforded by such models, a number of practical uses, such as the identification of suitable drug targets in cancer therapy, may be reaped by inferring the structure of the genetic models from experimental data, for example, from gene expression profiles (Huang [30]). To that end much recent work has gone into identifying the structure of gene regulatory networks from expression data (Liang et al. [48], Akutsu et al. [49, 50, 51], D'haeseleer et al. [52], Shmulevich et al. [53], Lähdesmäki et al. [54]). It is clear that "wet" lab experimental design and "dry" lab modeling and analysis must be tightly integrated and coordinated to generate, refine, validate, and interpret the biologically relevant models (see Figure 7.2).

Perhaps the most salient limitation of standard Boolean networks is their inherent determinism. From a conceptual point of view, it is likely that the regularity of genetic function and interaction known to exist is not due to "hard-wired" logical rules, but rather to the intrinsic self-organizing stability of the dynamical system, despite the existence of stochastic components in the cell. From an empirical point of view, there are two immediate reasons why the assumption of only one logical rule per gene may lead to incorrect conclusions when inferring these rules from gene expression measurements: (1) the measurements are typically noisy and the number of samples is small relative to the number of parameters to be inferred;

(2) the measurements may be taken under different conditions, and some rules may differ under these varying conditions.

7.3. Probabilistic Boolean networks

7.3.1. Background

The probabilistic Boolean network model was introduced by Shmulevich et al. [1]. These networks share the appealing properties of Boolean networks, but are able to cope with uncertainty, both in the data and the model selection. There are various reasons for utilizing a probabilistic network. A model incorporates only a partial description of a physical system. This means that a Boolean function giving the next state of a variable is likely to be only partially accurate. There will be conditions under which different Boolean functions may actually describe the transition, but these are outside the scope of the conventional Boolean model. If, consequently, we are uncertain as to which transition rule should be used, then a probabilistic Boolean network involving a set of possible Boolean functions for each variable may be more suitable than a network in which there is only a single function for each variable.

Even if one is fairly confident that a model is sufficiently robust that other variables can be ignored without significant impact, there remains the problem of inferring the Boolean functions from sample data. In the case of gene-expression microarrays, the data are severely limited relative to the number of variables in the system. Should it happen that a particular Boolean function has even a moderately large number of essential variables, its design from the data is likely to be imprecise because the number of possible input states will be too large for precise estimation. This situation is exacerbated if some essential variables are either unknown or unobservable (latent). As a consequence of the inability to observe sufficient examples to design the transition rule, it is necessary to restrict the number of variables over which a function is defined. For each subset of the full set of essential variables, there may be an optimal function, in the sense that the prediction error is minimized for that function, given the variables in the subset. These optimal functions must be designed from sample data. Owing to inherent imprecision in the design process, it may be prudent to allow a random selection between several functions, with the weight of selection based on a probabilistic measure of worth, such as the coefficient of determination (Dougherty et al. [55]).

The other basic issue regarding a probabilistic choice of transition rule is that in practice we are modeling an open system rather than a closed system. An open system has inputs (stimuli) that can affect regulation. Moreover, any model used to study a physical network as complex as a eukaryotic genome must inevitably omit the majority of genes. The states of those left out constitute external conditions to the model. System transition may depend on a particular external condition at a given moment of time. Such effects have been considered in the framework of using the coefficient of determination in the presence of external stresses (Kim et al. [56]). Under the assumption that the external stimuli occur asynchronously, it is prudent to allow uncertainty among the transition rules and

weight their likelihood accordingly. It may be that the probability of applying a Boolean function corresponding to an unlikely condition is low; however, system behavior might be seriously misunderstood if the possibility of such a transition is ignored.

It has been shown that Markov chain theory could be used to analyze the dynamics of PBNs (Shmulevich et al. [1]). Also, the relationships to Bayesian networks have been established, and the notions of *influences* and *sensitivities* of genes defined. The latter have been used to study the dynamics of Boolean networks (Shmulevich and Kauffman [57]). The inference of networks from gene expression data has received great attention. It is important for the inferred network to be robust in the face of uncertainty. Much work in this direction has already been carried out specifically in the context of gene regulatory networks (Dougherty et al. [58], Kim et al. [56, 59], Shmulevich et al. [53], Lähdesmäki et al. [54], Zhou et al. [10, 60, 61]). Visualization tools for the inferred multivariate gene relationships in networks are described by Suh et al. [62].

A framework for constructing subnetworks, adjoining new genes to subnetworks, and mapping between networks in such a way that the network structure and parameters remain consistent with the data has been established by Dougherty and Shmulevich [5]. An algorithm for growing subnetworks from so-called "seed genes" has been developed by Hashimoto et al. [11] and applied to several datasets (glioma and melanoma). An important goal of PBN modeling is to study the long-run behavior of the genetic networks. This was studied by Shmulevich et al. [6], using Markov chain Monte Carlo (MCMC) methods, along with a detailed analysis of convergence. In particular, the effect of network mappings on long-run behavior is critically important and a preliminary study has been carried out regarding the effect of network compression, including the issue of how to compress a network to reduce complexity while at the same time maintain long-run behavior to the extent possible (Ivanov and Dougherty [63]).

A gene perturbation model was extensively studied by Shmulevich et al. [2]. This approach not only simplified the steady-state analysis, but also provided a theoretical framework for assessing the effects of single-gene perturbations on the global long-run network behavior. In addition, a methodology for determining which genes would be good potential candidates for intervention was developed. Intervention and perturbation were presented in a unified framework. In addition, another approach for intervention, based on structural control with the use of genetic algorithms, was presented by Shmulevich et al. [3]. Finally, intervention based on external control was considered by Datta et al. [7, 8]. In that work, given a PBN whose state transition probabilities depend on an external (control) variable, a dynamic programming-based procedure was developed by which one could choose the sequence of control actions that minimized a given performance index over a finite number of steps.

7.3.2. Biological significance

One of the main objectives of Boolean-based network modeling is to study generic coarse-grained properties of large genetic networks and the logical interactions of

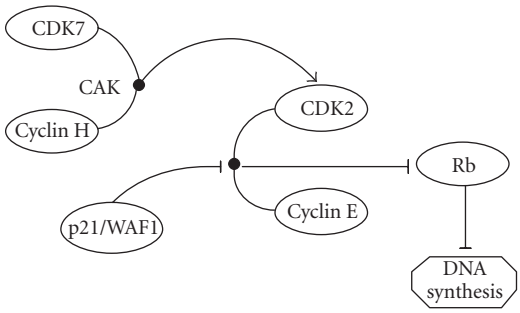


Figure 7.3. A diagram illustrating the cell cycle regulation example. Arrowed lines represent activation and lines with bars at the end represent inhibition.

genes, without knowing specific quantitative biochemical details, such as kinetic parameters of individual reactions. The biological basis for the development of Boolean networks as models of genetic regulatory networks lies in the fact that during regulation of functional states, the cell exhibits switch-like behavior, which is important for cells to move from one state to another in a normal cell growth process or in situations when cells need to respond to external signals, many of which are detrimental. Let us use cell cycle regulation as an example. Cells grow and divide. This process is highly regulated; failure to do so results in unregulated cell growth in diseases such as cancer. In order for cells to move from the G1 phase to the S phase, when the genetic material, DNA, is replicated for the daughter cells, a series of molecules such as cyclin E and cyclin-dependent kinase 2 (CDK2) work together to phosphorylate the retinoblastoma (Rb) protein and inactivate it, thus releasing cells into the S phase. CDK2/cyclin E is regulated by two switches: the positive switch complex called CDK activating kinase (CAK) and the negative switch p21/WAF1. The CAK complex can be composed of two gene products: cyclin H and CDK7. When cyclin H and CDK7 are present, the complex can activate CDK2/cyclin E. A negative regulator of CDK2/cyclin E is p21/WAF1, which in turn can be activated by p53. When p21/WAF1 binds to CDK2/cyclin E, the kinase complex is turned off (Gartel and Tyner [64]). Further, p53 can inhibit cyclin H, a positive regulator of cyclin E/CDK2 (Schneider et al. [65]). This negative regulation is an important defensive system in the cells. For example, when cells are exposed to mutagens, DNA damage occurs. It is to the benefit of cells to repair the damage before DNA replication so that the damaged genetic materials do not pass onto the next generation. Extensive amount of work has demonstrated that DNA damage triggers switches that turn on p53, which then turns on p21/WAF1. p21/WAF1 then inhibits CDK2/cyclin E, thus Rb becomes activated and DNA synthesis stops. As an extra measure, p53 also inhibits cyclin H, thus turning off the switch that turns on CDK2/cyclin E. Such delicate genetic switch networks in the cells are the basis for cellular homeostasis.

For purposes of illustration, let us consider a simplified diagram, shown in Figure 7.3, illustrating the effects of CDK7/cyclin H, CDK2/cyclin E, and p21/WAF1 on Rb. Thus, p53 and other known regulatory factors are not considered.

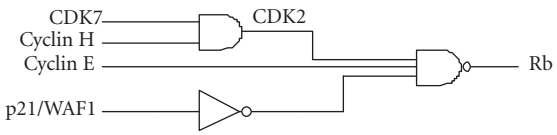


Figure 7.4. The logic diagram describing the activity of retinoblastoma (Rb) protein in terms of 4 inputs: CDK7, cyclin H, cyclin E, and p21. The gate with inputs CDK7 and cyclin H is an AND gate, the gate with input p21/WAF1 is a NOT gate, and the gate whose output is Rb is a NAND (negated AND) gate.

While this diagram represents the above relationships from a pathway perspective, we may also wish to represent the activity of Rb in terms of the other variables in a logic-based fashion. Figure 7.4 contains a logic circuit diagram of the activity of Rb (on or off) as a Boolean function of four input variables: CDK7, cyclin H, cyclin E, and p21/WAF1. Note that CDK2 is shown to be completely determined by the values of CDK7 and cyclin H using the AND operation and thus, CDK2 is not an independent input variable. Also, in Figure 7.3, p21/WAF1 is shown to have an inhibitive effect on the CDK2/cyclin E complex, which in turn regulates Rb, while in Figure 7.4, we see that from a logic-based perspective, the value of p21/WAF1 works together with CDK2 and cyclin E to determine the value of Rb. Such dual representations in the biological literature were pointed out by Rzhetsky et al. [66].

7.3.3. Definitions

Mathematically, a Boolean network $G(V,F)$ is defined by a set of nodes $V = \{x_1,\ldots,x_n\}$ and a list of Boolean functions $F = \{f_1,\ldots,f_n\}$. Each x_i represents the state (expression) of a gene i, where $x_i = 1$ represents the fact that gene i is expressed and $x_i = 0$ means it is not expressed. It is commonplace to refer to $x_1, x_2, \ldots, x_n$ as genes. The list of Boolean functions F represents the rules of regulatory interactions between genes. Each $x_i \in \{0,1\}$, $i = 1,\ldots,n$, is a binary variable and its value at time $t+1$ is completely determined by the values of some other genes $x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)}$ at time t by means of a Boolean function $f_i \in F$. That is, there are k_i genes assigned to gene x_i and the set $\{x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)}\}$ determines the "wiring" of gene x_i. Thus, we can write

$$x_i(t+1) = f_i\left(x_{j_1(i)}(t), x_{j_2(i)}(t), \ldots, x_{j_{k_i}(i)}(t)\right). \tag{7.1}$$

The *maximum connectivity* of a Boolean network is defined by $K = \max_i k_i$. All genes are assumed to update synchronously in accordance with the functions assigned to them and this process is then repeated. The artificial synchrony simplifies computation while preserving the qualitative, generic properties of global network dynamics (Huang [30], Kauffman [31], Wuensche [46]). It is clear that the dynamics of the network are completely deterministic.

The basic idea behind PBNs is to combine several promising Boolean functions, now called *predictors*, so that each can make a contribution to the prediction of a target gene. A natural approach is to allow a random selection of the

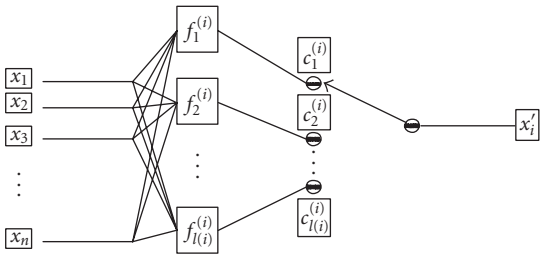


Figure 7.5. A basic building block of a probabilistic Boolean network. Although the "wiring" of the inputs to each function is shown to be quite general, in practice, each function (predictor) has only a few input variables.

predictors for a given target gene, with the selection probability being proportional to some measure of the predictor's determinative potential, such as the coefficient of determination, described later. At this point, it suffices for us to assume that each predictor has an associated probability of being selected. Given genes $V = \{x_1, \ldots, x_n\}$, we assign to each x_i a set $F_i = \{f_1^{(i)}, \ldots, f_{l(i)}^{(i)}\}$ of Boolean functions composed of the predictors for that target gene. Clearly, if $l(i) = 1$ for all $i = 1, \ldots, n$, then the PBN simply reduces to a standard Boolean network. The basic building block of a PBN is shown in Figure 7.5.

As first introduced in Shmulevich et al. [1], at each point in time or step of the network, a function $f_j^{(i)}$ is chosen with probability $c_j^{(i)}$ to predict gene x_i. Considering the network as a whole, a *realization* of the PBN at a given instant of time is determined by a vector of Boolean functions, where the ith element of that vector contains the predictor selected at that instant for gene x_i. If there are N possible realizations, then there are N vector functions, $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N$, of the form $f_k = (f_{k_1}^{(1)}, f_{k_2}^{(2)}, \ldots, f_{k_n}^{(n)})$, for $k = 1, 2, \ldots, N$, $1 \le k_i \le l(i)$, and where $f_{k_i}^{(i)} \in F_i$ $(i = 1, \ldots, n)$. In other words, the vector function $\mathbf{f}_k : \{0,1\}^n \to \{0,1\}^n$ acts as a transition function (mapping) representing a possible realization of the entire PBN. Such functions are commonly referred to as multiple-output Boolean functions. In the context of PBNs we refer to them as *network functions*. If we assume that the predictor for each gene is chosen independently of other predictors, then $N = \prod_{i=1}^{n} l(i)$. More complicated dependent selections are also possible. Each of the N possible realizations can be thought of as a standard Boolean network that operates for one time step. In other words, at every state $x(t) \in \{0,1\}^n$, one of the N Boolean networks is chosen and used to make the transition to the next state $x(t+1) \in \{0,1\}^n$. The probability P_i that the ith (Boolean) network or realization is selected can be easily expressed in terms of the individual selection probabilities $c_j^{(i)}$ (see Shmulevich et al. [1]).

The PBN model is generalized by assuming that the decision to select a new network realization is made with probability λ at every time step. In other words, at every time step, a coin is tossed with probability λ of falling on heads, and if it does, then a new network realization is selected as described above; otherwise, the current network realization is used for the next time step. The original PBN

definition as described above corresponds to the case $\lambda = 1$. We will refer to the model with $\lambda = 1$ as an *instantaneously random* PBN. The $\lambda < 1$ has a natural interpretation relative to external conditions. The Boolean network remains unchanged from moment to moment, except when its regulatory structure is altered by a change in an external condition. Any given set of conditions may be considered to correspond to a context of the cell. Hence, when $\lambda < 1$ we refer to the network as a *context-sensitive* PBN. Assuming conditions are stable, λ will tend to be quite small (Braga-Neto et al. [67], Zhou et al. [61], Brun et al. [68]).

Thus far, randomness has only been introduced relative to the functions (hence, implicitly, also the connectivity); however the model can be extended to incorporate transient gene perturbations. Suppose that a gene can get perturbed with (a small) probability p, independently of other genes. In the Boolean setting, this is represented by a flip of value from 1 to 0 or vice versa. This type of "randomization," namely, allowing genes to randomly flip value, is biologically meaningful. Since the genome is not a closed system, but rather has inputs from the outside, it is known that genes may become either activated or inhibited due to external stimuli, such as mutagens, heat stress, and so forth. Thus, a network model should be able to capture this phenomenon.

Suppose that at every step of the network we have a realization of a random *perturbation vector* $\gamma \in \{0,1\}^n$. If the ith component of γ is equal to 1, then the ith gene is flipped, otherwise it is not. In general, γ need not be independent and identically distributed (i.i.d.), but will be assumed so for simplicity. Thus, we will suppose that $\Pr\{\gamma_i = 1\} = E[\gamma_i] = p$ for all $i = 1,\ldots,n$. Let $x(t) \in \{0,1\}^n$ be the state of the network at time t. Then, the next state $x(t+1)$ is given by

$$x(t+1) = \begin{cases} x(t) \oplus \gamma, & \text{with probability } 1-(1-p)^n, \\ \mathbf{f}_k\left(x_1(t),\ldots,x_n(t)\right), & \text{with probability } (1-p)^n, \end{cases} \tag{7.2}$$

where $\oplus$ is componentwise addition modulo 2 and $\mathbf{f}_k$, $k = 1,2,\ldots,N$, is the network function representing a possible realization of the entire PBN, as discussed above.

For a PBN with random perturbation, the following events can occur at any point of time: (1) the current network function is applied, the PBN transitions accordingly, and the network function remains the same for the next transition; (2) the current network function is applied, the PBN transitions accordingly, and a new network function is selected for the next transition; (3) there is a random perturbation and the network function remains the same for the next transition; (4) there is a random perturbation and a new network function is selected for the next transition.

7.4. Long-run behavior

7.4.1. Steady-state distribution

In the absence of random perturbations, the states of an instantaneously random PBN form a finite-state homogeneous Markov chain and that possesses a

stationary distribution, where the transition probabilities depend on the network functions. When random perturbations are incorporated into the model, the chain becomes ergodic and possesses a steady-state distribution. In Shmulevich et al. [2], an explicit formulation of the state-transition probabilities of the Markov chain associated with the PBN is derived in terms of the Boolean functions and the probability of perturbation p.

In the case of a context-sensitive PBN, the state vector of gene values at time t cannot be considered as a homogeneous Markov chain anymore because the transition probabilities depend on the function selected at time t. Instead of representing the states $\mathbf{x}$ of the PBN as the states of a Markov chain, we can represent the state-function pairs $(\mathbf{x}, \mathbf{f}_k)$ as states of a homogeneous Markov chain with transition probabilities

$$P_{\mathbf{y},\mathbf{f}_l}(\mathbf{x}, \mathbf{f}_k) = P(\mathbf{X}_t = \mathbf{x},\ \mathbf{F}_t = \mathbf{f}_k \mid \mathbf{X}_{t-1} = \mathbf{y},\ \mathbf{F}_{t-1} = \mathbf{f}_l) \tag{7.3}$$

for any time t. The chain must possess a stationary distribution, and if there are random perturbations, then it possesses a steady-state distribution. The probabilities $\pi(\mathbf{x})$ are the marginal probabilities of the steady-state distribution defined by

$$\pi(\mathbf{x}, \mathbf{f}_k) = \lim_{t\to\infty} P(\mathbf{X}_{t_0+t} = \mathbf{x},\ \mathbf{F}_{t_0+t} = \mathbf{f}_k \mid \mathbf{X}_{t_0} = \mathbf{y},\ \mathbf{F}_{t_0} = \mathbf{f}_l), \tag{7.4}$$

where t_0 is the initial time. These steady-state distributions for context-sensitive PBNs have been studied by Brun et al. [68].

7.4.2. Attractors

Owing to its deterministic and finite nature, if a Boolean network is initialized and then allowed to dynamically transition, it will return to a previously visited state within a bounded amount of time (based on the total number of genes). Once this occurs, it will cycle from that state through the same set of states and in the same order as it did after following the first visit to the state. The cycle of states is called an attractor cycle. Note that attractor cycles must be disjoint and either every state is a member of an attractor or it is transient, meaning it cannot be visited more than once. Each initialization leads to a unique attractor and the set of states leading to an attractor is called the basin of attraction for the attractor. A singleton attractor (absorbing state) has the property that once entered, the network cannot leave it.

The attractors of a Boolean network characterize its long-run behavior. If, however, we incorporate random perturbation, then the network can escape its attractors. In this case, full long-run behavior is characterized by its steady-state distribution. Nonetheless, if the probability of perturbation is very small, the network will lie in its attractor cycles for a large majority of the time, meaning that attractor states will carry most of the steady-state probability mass. The amount of time spent in any given attractor depends on its basin. Large basins tend to produce attractors possessing relatively large steady-state mass.

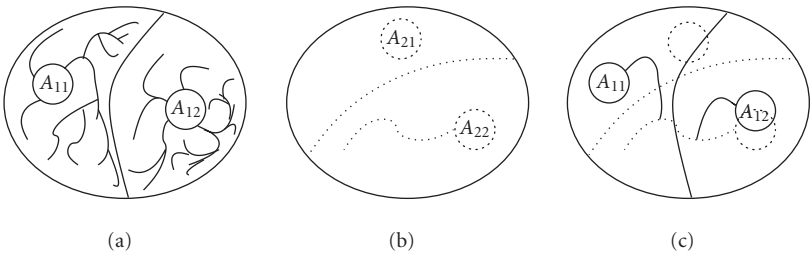


Figure 7.6. An illustration of the behavior of a context-sensitive PBN.

Let us now consider context-sensitive PBNs. So long as there is no switching, the current Boolean-network realization of the PBN characterizes the activity of the PBN and it will transition into one of its attractor cycles and remain there until a switch occurs. When a network switch does occur, the present state becomes an initialization for the new realization and the network will transition into the attractor cycle whose basin contains the state. It will remain there until another network switch. The attractor family of the PBN is defined to be the union of all the attractors in the constituent Boolean networks. Note that the attractors of a PBN need not be disjoint, although those corresponding to each constituent Boolean network must be disjoint.

Figure 7.6 shows an example of the behavior of a context-sensitive PBN relative to its attractors under a change of function. Part (a) shows the attractor cycles A_{11} and A_{12} for a network function $\mathbf{f}_1$, its corresponding basins, and some trajectories. Part (b) shows the attractor cycles A_{21} and A_{22} for a network function $\mathbf{f}_2$ and its corresponding basins. In part (c), we can see that if the system is using the function $\mathbf{f}_2$ and it makes a function change, to $\mathbf{f}_1$, then the future of the system depends on which part of the trajectory it is at the moment of the function change. In this example, for the particular trajectory shown with the dotted line toward the attractor A_{22}, the first part of the trajectory is in the basin corresponding to the attractor A_{11}, and the end of the trajectory is inside the basin corresponding to the attractor A_{12}. Therefore, if the change of function occurs before the system crosses the boundary between the basins, it will transition toward the attractor A_{11}. If the change of function occurs after it crosses the boundary, then it will transition toward the attractor A_{12}. In particular, we see that the attractor A_{22} lies completely inside the basin corresponding to the attractor A_{12}. In this case, if a change of function occurs when the system is inside the attractor A_{22}, it will always transition to the attractor A_{12}.

If one now incorporates perturbation into the PBN model, the stationary distribution characterizes the long-run behavior of the network. If both the switching and perturbation probabilities are very small, then the attractors still carry most of the steady-state probability mass. This property has been used to formulate analytic expressions of the probabilities of attractor states (Brun et al. [68]) and to validate network inference from data (Kim et al. [9], Zhou et al. [61]).

7.4.3. Monte-Carlo estimation of the steady-state distribution

Model-based simulations are invaluable for gaining insight into the underlying functioning of a genetic regulatory network. Simulation-supported decision making is essential in realistic analysis of complex dynamical systems. For example, one may wish to know the long-term joint behavior of a certain group of genes or the long-term effect of one gene on a group of others. After having robustly inferred the model structure and parameters, such questions can be answered by means of simulations. We have developed a methodology for analyzing steady-state (or long-run) behavior of PBNs using MCMC-type approaches (Shmulevich et al. [6]). By simulating the network until it converges to its steady-state distribution and monitoring the convergence by means of various diagnostics (Cowles and Carlin [69]), we can obtain the limiting probabilities of the genes of interest. Thus, the effects of permanent and transient interventions (e.g., turning a gene off) can be assessed on the long-run network behavior.

An approach found to be useful for determining the number of iterations necessary for convergence to the stationary distribution of the PBN is based on a method by Raftery and Lewis [70]. This method reduces the study of the convergence of the Markov chain corresponding to a PBN to the study of the convergence of a two-state Markov chain. Suppose that we are interested in knowing the steady-state probability of the event {Gene A is ON and Gene B is OFF}. Then, we can partition the state space into two disjoint subsets such that one subset contains all states on which the event occurs and the other subset contains the rest of the states. Consider the two meta-states corresponding to these two subsets. Although the sequence of these meta-states does not form a Markov chain in itself, it can be approximated by a first-order Markov chain if every k states from the original Markov chain is discarded (i.e., the chain is subsampled). It turns out in practice that k is usually equal to 1, meaning that nothing is discarded and the sequence of meta-states is treated as a homogeneous Markov chain (see Raftery and Lewis for details) with transition probabilities α and β between the two meta-states. Using standard results for two-state Markov chains, it can be shown that the burn-in period (the number of iterations necessary to achieve stationarity) m_0 satisfies

$$m_0 \geq \frac{\log\left(\varepsilon(\alpha+\beta)/\max(\alpha,\beta)\right)}{\log(1-\alpha-\beta)}. \tag{7.5}$$

We set $\varepsilon = 0.001$ in our experiments. In addition, it can be shown that the minimum total number of iterations N necessary to achieve a desired accuracy r (we used $r = 0.01$ in our experiments) is

$$N = \frac{\alpha\beta(2-\alpha-\beta)}{(\alpha+\beta)^3}\left(\frac{r}{\Phi\left((1/2)(1+s)\right)}\right)^{-2}, \tag{7.6}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function and s is a parameter that we set to 0.95 in our experiments. For detailed explanations of the precision parameters ε, r, and s, see Raftery and Lewis [70]. The question becomes

Table 7.1. An example of the joint steady-state probabilities (in percentages) of several pairs of genes, computed from the network inferred from glioma gene expression data.

Tie-2	NFκB	%	Tie-2	TGFB3	%	TGFB3	NFκB	%
Off	Off	15.68	Off	Off	14.75	Off	Off	10.25
Off	On	41.58	Off	On	42.50	Off	On	12.47
On	Off	9.21	On	Off	7.96	On	Off	14.64
On	On	31.53	On	On	32.78	On	On	60.65

how to estimate the transition probabilities α and β, as these are unknown. The solution is to perform a test run from which α and β can be estimated and from which m_0 and N can be computed. Then, another run with the computed burn-in period m_0 and the total number of iterations N is performed and the parameters α and β are reestimated from which m_0 and N are recomputed. This can be done several times in an iterative manner until the estimates of m_0 and N are smaller than the number of iterations already achieved. We have used this method to determine the steady-state probabilities of some genes of interest from our gene expression data set, as described below.

We analyzed the joint steady-state probabilities of several combinations of two genes from a subnetwork generated from our glioma expression data: Tie-2 and NFκB, Tie-2 and TGFB3, and TGFB3 and NFκB. The steady-state probabilities for all pairs of considered genes are shown in Table 7.1 as percentages. Tie-2 is a receptor tyrosine kinase expressed on the endothelial cells. Its two ligands, angiopoietins 1 and 2, bind Tie-2 and regulate vasculogenesis (Sato et al. [71]), an important process in embryonic development and tumor development. Other related regulators for vasculogenesis are VEGF and VERF receptors, which are often overexpressed in the advanced stage of gliomas (Cheng et al. [72]). Although no experimental evidence supports a direct transcriptional regulation of those regulators by the transcriptional factor NFκB, which is also frequently activated in glioma progression (Hayashi et al. [73]) as predicted in this analysis, the results show that NFκB, at least indirectly, influence the expression of Tie-2 expression. Thus, it may not be surprising that when NFκB is on, Tie-2 is on about $31.53/(41.58 + 31.53) = 43\%$ of time. Because Tie-2 is only one of the regulators of vasculogenesis, which is important in glioma progression, it is consistent that our analysis of long-term (steady-state) gene expression activities shows that about 40% of the time Tie-2 is on. In contrast, NFκB is on 73% of the time, implying that fewer redundancies exist for NFκB activity.

Interestingly, a similar relationship exists between Tie-2 and TGFB3, as can be seen by comparing the percentages in columns 3 and 6 of Table 7.1. This suggests that TGFB3 and NFκB are more directly linked, which is also shown in the last three columns of the table (60% of the time, they are both on). This relationship is supported by the fact that TGFB1, a homologue of TGFB3, was shown to have a direct regulatory relationship with NFκB (Arsura et al. [74]) as well as by the recent work of Strauch et al. [75], who recently showed that NFκB activation indeed upregulates TGFB expression.

7.5. Inference of PBNs from gene expression data

Owing to several current limitations, such as availability of only transcriptional measurements (much regulation occurs on the protein level), cell population asynchrony and possible heterogeneity (different cell types exhibiting different gene activities), and latent factors (environmental conditions, genes that we are not measuring, etc.), it is prudent to strive to extract higher-level information or knowledge about the relationships between measurements of gene transcript levels. If we can discover such relationships, then we can potentially learn something new about the underlying mechanisms responsible for generating these observed quantities. The burden of discovery remains in the wet lab, where potentially interesting relationships must be examined experimentally. We now discuss some approaches to the inference problem. These approaches have already been used in a number of different studies. At this point it still remains to be known which aspects of the data tend to be emphasized by which approach and whether one approach reflects the actual regulatory relationships more faithfully than another. This constitutes an important research problem.

7.5.1. Coefficient of determination

A basic building block of a rule-based network is a *predictor*. In a probabilistic network, several good predictors are probabilistically synthesized to determine the activity of a particular gene. A predictor is designed from data, which means that it is an approximation of the predictor whose action one would actually like to model. The precision of the approximation depends on the design procedure and the sample size.

Even in the context of limited data, modest approaches can be taken. One general statistical approach is to discover associations between the expression patterns of genes via the *coefficient of determination* (CoD) (Dougherty et al. [55, 58], Kim et al. [56, 59]). This coefficient measures the degree to which the transcriptional levels of an observed gene set can be used to improve the prediction of the transcriptional state of a target gene relative to the best possible prediction in the absence of observations. Let Y be a target variable, $\mathbf{X}$ a set of variables, and f the function such that $f(\mathbf{X})$ is the optimal predictor of Y relative to minimum mean-square error, $\varepsilon(Y, f(\mathbf{X}))$. The CoD for Y relative to $\mathbf{X}$ is defined by

$$\theta_{\mathbf{X}}(Y) = \frac{\varepsilon_{\bullet}(Y) - \varepsilon\big(Y, f(\mathbf{X})\big)}{\varepsilon_{\bullet}(Y)}, \tag{7.7}$$

where $\varepsilon_{\bullet}(Y)$ is the error of the best constant estimate of Y in the absence of any conditional variables. The CoD is between 0 and 1.

The method allows incorporation of knowledge of other conditions relevant to the prediction, such as the application of particular stimuli, or the presence of inactivating gene mutations, as predictive elements affecting the expression level of a given gene. Using the coefficient of determination, one can find sets of genes

related multivariately to a given target gene. The CoD has been used in gene-expression studies involving genotoxic stress (Kim et al. [56]), melanoma (Kim et al. [9]), glioma (Hashimoto et al. [11]), and atherogenesis (Johnson et al. [76]).

The coefficient of determination is defined in terms of the population distribution. However, in practice, we use the sample-based version; much like the sample mean (average) is the estimate of the population mean. An important research goal related to the CoD is to study and characterize the behavior of its estimator in the context of robustness. That is, it is important to understand to what extent the presence of outliers influences the estimate of the CoD. Various tools to analyze robustness, used in the nonlinear signal processing community (e.g., Shmulevich et al. [77]), may be applicable in this context.

7.5.2. Best-fit extensions

Most recent work with Boolean networks has focused on identifying the structure of the underlying gene regulatory network from gene expression data (Liang et al. [48], Akutsu et al. [49, 50], Ideker et al. [78], Karp et al. [79], Maki et al. [80], Noda et al. [81], Shmulevich et al. [53]). A related issue is to find a network that is consistent with the given observations or determine whether such a network exists at all. Much work in this direction has been traditionally focused on the so-called *consistency problem*, namely, the problem of determining whether or not there exists a network that is consistent with the observations.

The consistency problem represents a search for a rule from examples. That is, given some sets T and F of true and false vectors, respectively, the aim is to discover a Boolean function f that takes on the value 1 for all vectors in T and the value 0 for all vectors in F. It is also commonly assumed that the target function f is chosen from some class of possible target functions. In the context of Boolean networks, such a class could be the class of canalizing functions (discussed later) or functions with a limited number of essential variables. More formally, let $T(f) = \{v \in \{0,1\}^n : f(v) = 1\}$ be called the *on-set* of function f and let $F(f) = \{v \in \{0,1\}^n : f(v) = 0\}$ be the *off-set* of f. The sets $T, F \subseteq \{0,1\}^n$, $T \cap F = \emptyset$, define a *partially defined* Boolean function $g_{T,F}$ as

$$g_{T,F}(v) = \begin{cases} 1, & v \in T \\ 0, & v \in F \\ *, & \text{otherwise.} \end{cases} \tag{7.8}$$

A function f is called an *extension* of $g_{T,F}$ if $T \subseteq T(f)$ and $F \subseteq F(f)$. The consistency problem (also called the extension problem) can be posed as follows: given a class C of functions and two sets T and F, is there an extension $f \in C$ of $g_{T,F}$?

While this problem is important in computational learning theory, since it can be used to prove the hardness of learning for various function classes (e.g., Shmulevich et al. [82]), it may not be applicable in realistic situations containing

noisy observations, as is the case with microarrays. That is, due to the complex measurement process, ranging from hybridization conditions to image processing techniques, as well as actual biological variability, expression patterns exhibit uncertainty.

A learning paradigm that can incorporate such inconsistencies is called the best-fit extension problem. Its goal is to establish a network that would make as few misclassifications as possible. The problem is formulated as follows. Suppose we are given positive weights $w(x)$ for all vectors $x \in T \cup F$ and define $w(S) = \sum_{x \in S} w(x)$ for a subset $S \subseteq T \cup F$. Then, the *error size* of function f is defined as

$$\varepsilon(f) = w(T \cap F(f)) + w(F \cap T(f)). \tag{7.9}$$

If $w(x) = 1$ for all $x \in T \cup F$, then the error size is just the number of misclassifications. The goal is then to output subsets T^* and F^* such that $T^* \cap F^* = \varnothing$ and $T^* \cup F^* = T \cup F$ for which the partially defined Boolean function g_{T^*,F^*} has an extension in some class of functions C and so that $w(T^* \cap F) + w(F^* \cap T)$ is minimum. Consequently, any extension $f \in C$ of g_{T^*,F^*} has minimum error size. A crucial consideration is computational complexity of the learning algorithms. In order for an inferential algorithm to be useful, it must be computationally tractable. It is clear that the best-fit extension problem is computationally more difficult than the consistency problem, since the latter is a special case of the former, that is, when $\varepsilon(f) = 0$. Shmulevich et al. [53] showed that, for many function classes, including the class of all Boolean functions, the best-fit extension problem is polynomial-time solvable in the number of genes and observations, implying its practical applicability to real data analysis. Also, fast optimized and scalable search algorithms for best-fit extensions were developed by Lähdesmäki et al. [54].

The best-fit method is very versatile in the sense that one can specify the relative cost of making an error in the inference for various states of gene expression. There are a number of available quality measurements (Chen et al. [83]) which could be used in this fashion. Thus, instead of discarding low-quality measurements, one may be able to control their relative influence by down-weighting them in the best-fit extension inference method, in proportion to their quality measure. This is a useful topic to explore in future research.

7.5.3. Bayesian connectivity-based design

A recently proposed Bayesian method for constructing PBNs (that applies to a more general class of networks) is based on the network connectivity scheme (Zhou et al. [61]). Using a reversible-jump MCMC technique, the procedure finds possible regulatory gene sets for each gene, the corresponding predictors, and the associated probabilities based on a neural network with a very simple hidden layer. An MCMC method is used to search the network configurations to find those with the highest Bayesian scores from which to construct the PBN. We briefly outline the method, leaving the details to the original paper.

Consider a Boolean network $G(V,F)$ as previously defined, V being the genes and F the predictors. Construction is approached in a Bayesian framework relative to network topology by searching for networks with the highest a posteriori probabilities

$$P(V|D) \propto P(D|V)P(V), \tag{7.10}$$

where D is the data set and $P(V)$, the prior probability for the network, is assumed to satisfy a uniform distribution over all topologies. Note that $P(V|D)$ is given by

$$P(V|D) = \int p(D|V,F)\,p(F)dF, \tag{7.11}$$

where p denotes the density. The computation of this integral is virtually intractable and therefore it is approximated (Zhou et al. [61]).

In the context of this Bayesian framework, a PBN is constructed by searching the space of network topologies and selecting those with the highest Bayesian scores $P(V|D)$ to form the PBN. The algorithm proceeds in the following general manner: generate an initial graph $V^{(0)}$; compute $P(V|D)$; for $j = 1, 2, \ldots$, calculate the predictors $F^{(j)}$ corresponding to $G^{(j)}$; compute the Bayesian $P(V|D)$ score; and choose $G^{(j+1)}$ via an MCMC step.

7.5.4. Plausible classes of genetic interactions

While the focus in computational learning theory has mostly been on the complexity of learning, very similar types of problems have been studied in nonlinear signal processing, specifically, in optimal filter design (Coyle and Lin [84], Coyle et al. [85], Yang et al. [86], Dougherty and Loce [87], Dougherty and Chen [88]). This typically involves designing an estimator from some predefined class of estimators that minimizes the error of estimation among all estimators in the class. An important role in filter design is played by these predefined classes or constraints. For example, the so-called stack filters are represented by the class of monotone Boolean functions. Although it would seem that imposing such constraints can only result in a degradation of the performance (larger error) relative to the optimal filter with no imposed constraints, constraining may confer certain advantages. These include prior knowledge of the degradation process (or in the case of gene regulatory networks, knowledge of the likely class of functions, such as canalizing functions), tractability of the filter design, and precision of the estimation procedure by which the optimal filter is estimated from observations. For example, we often know that a certain class of filters will provide a very good suboptimal filter, while considerably lessening the data requirements for its estimation. It is with the issue of filter complexity versus sample size that the design of nonlinear filters intersects with the theory of classification (Dougherty and Barrera [89]). We now discuss several promising constraints for inferring Boolean predictors for PBNs.

An important class of functions, known to play an important role in regulatory networks (Kauffman [31, 90], Harris et al. [91]), is the class of *canalizing functions*. Canalizing functions constitute a special type of Boolean function in which at least one of the input variables is able to determine the value of the output of the function. For example, the function $f(x_1, x_2, x_3) = x_1 + x_2x_3$, where the addition symbol stands for disjunction and the multiplication for conjunction, is a canalizing function, since setting x_1 to 1 guarantees that the value of the function becomes 1 regardless of the value of x_2 or x_3. Although their defining property is quite simple, canalizing functions have been implicated in a number of phenomena related to discrete dynamical systems as well as nonlinear digital filters (see references in Shmulevich et al. [92]).

Canalizing functions, when used as regulatory control rules, are one of the few known mechanisms capable of preventing chaotic behavior in Boolean networks (Kauffman [31]). In fact, there is overwhelming evidence that canalizing functions are abundantly utilized in higher vertebrate gene regulatory systems. Indeed, a recent large-scale study of the literature on transcriptional regulation in eukaryotes demonstrated an overwhelming bias towards canalizing rules (Harris et al. [91]).

Recently, Shmulevich et al. [93] have shown that certain *Post classes*, which are classes of Boolean functions that are closed under superposition (Post [94]), also represent plausible evolutionarily selected candidates for regulatory rules in genetic networks. These classes have also been studied in the context of synthesis (Nechiporuk [95]) and reliability (Muchnik and Gindikin [96]) of control systems—a field that bears an important relationship to genetic control networks. The Post classes considered by Shmulevich et al. [93] play an important role in the emergence of order in Boolean networks. The closure property mentioned above implies that any gene at any number of steps in the future is guaranteed to be governed by a function from the same class. It was demonstrated that networks constructed from functions belonging to these classes have a tendency toward ordered behavior and are not overly sensitive to initial conditions, moreover and damage does not readily spread throughout the network. In addition, the considered classes are significantly larger than the class of canalizing functions, as the number of inputs per Boolean function increases. Additionally, functions from this class have a natural way to ensure robustness against noise and perturbations, thus representing plausible evolutionarily selected candidates for regulatory rules in genetic networks. Efficient spectral algorithms for testing membership of functions in these classes as well as the class of canalizing functions have been developed by Shmulevich et al. [92].

An important research goal is to determine whether the constraints described above are plausible not only from the point of view of evolution, noise resilience, and network dynamical behavior, but also in light of experimental data. Tools from model selection theory can be used to answer this question. Should this prove to be the case, by having prior knowledge of the plausible rules of genetic interaction, one can significantly improve model inference by reducing data requirements and increasing accuracy and robustness.

7.6. Subnetworks

It is likely that genetic regulatory networks function in what might be called a *multiscale* manner. One of the basic principles in multiscale modeling is that meaningful and useful information about a system or object exists on several different "levels" simultaneously. In the context of genetic networks, this would imply that genes form small groups (or clusters) wherein genes have close interactions. Some of these clusters are functionally linked forming larger "metaclusters" and these metaclusters have interactions as well. This process may continue on several different scales. This type of clustering effect has been observed in many other types of networks, such as social networks (Newman et al. [97]), the power grid of the western United States, and neural networks (Watts and Strogatz [98]). Interestingly, dynamical systems that have this property exhibit enhanced signal-propagation speed and computational power.

7.6.1. Growing subnetworks from seed genes

An important goal is to discover relatively small subnetworks, out of the larger overall network, that function more or less independently of the rest of the network. Such a small subnetwork would require little or sometimes even no information from the "outside." We can proceed by starting with a "seed" consisting of one or more genes that are believed to participate in such a subnetwork. Then, we iteratively adjoin new genes to this subnetwork such that we maintain the aforementioned "autonomy" of the subnetwork as much as possible, using the notions of gene influence (Shmulevich et al. [1]) or the coefficient of determination. Such an algorithm for growing subnetworks from seed genes has been developed by Hashimoto et al. [11].

Subnetwork construction proceeds in a way that enhances a strong collective strength of connections among the genes within the subnetwork and also limits the collective strength of the connections from outside the subnetwork. Consider Figure 7.7. Suppose we have a subnetwork $\mathbf{S}$ and are considering the candidate gene Y for inclusion in this subnetwork. We would like the collective strength (to be defined in a moment) of the genes in $\mathbf{S}$ on the candidate gene Y as well as the strength of gene Y on the genes in $\mathbf{S}$ to be high. In other words, the genes in $\mathbf{S}$ and Y should be tightly interdependent. At the same time, other genes from outside of the subnetwork should have little impact on Y if we are to maintain the subnetwork autonomy or "self determinacy." Thus, their collective strength on Y should be low. At each step, the subnetwork grows by one new gene so as to ensure maximal autonomy. An overall measure of subnetwork autonomy, which serves as an objective function in the subnetwork growing algorithm, is a combination of the three types of strength just described (Hashimoto et al. [11]). Finally, the strength itself can be naturally captured either by the coefficient of determination or by the influence, which we now discuss.

The *influence* of a variable relative to a Boolean function for which it is one among several Boolean variables is defined via the partial derivative of a Boolean

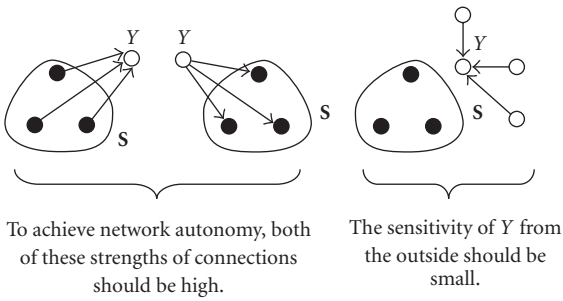


Figure 7.7. In order to maintain the self-autonomy of subnetwork **S**, the collective strength of the genes in **S** on gene Y—a candidate for inclusion in the subnetwork—should be high. The strength of Y on **S** should be high as well, thus maintaining high interdependency between the genes in the subnetwork. At the same time, the strength of genes outside the subnetwork on gene Y should be low.

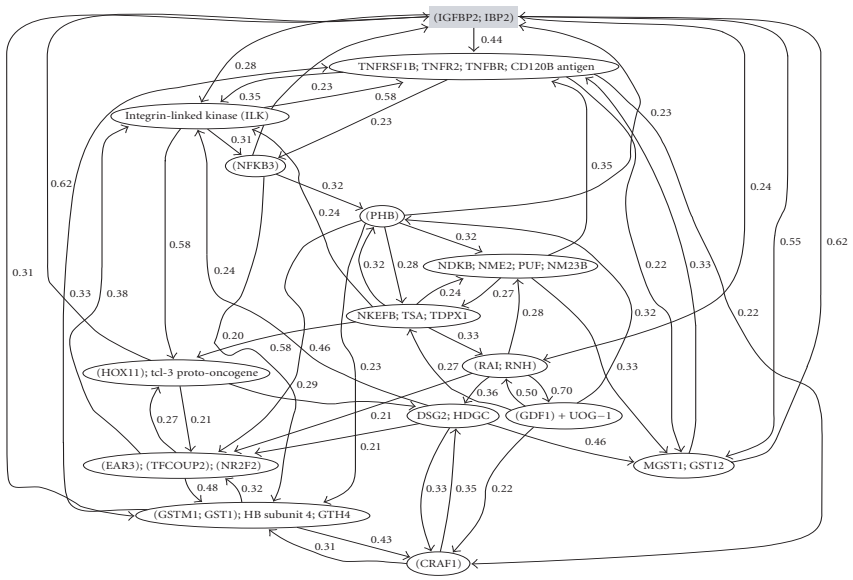
function. One can define the partial derivative of a Boolean function in several equivalent ways; however, for our purposes here, we simply note that the partial derivative of f with respect to the variable x_j is 0 if toggling the value of variable x_j does not change the value of the function, and it is 1 otherwise. The influence of x_j on f is the expectation of the partial derivative with respect to the distribution of the variables. In the context of a probabilistic Boolean network, there are a number of predictor functions associated with each gene, and each of these functions has associated with it a selection probability (Shmulevich et al. [1]). The influence of gene x_k on gene x_j is the sum of the influences of gene x_k on x_j relative to the family of predictor functions for x_j, weighted by the selection probabilities for these x_j-predicting functions.

Examples of subnetworks with IGFBP2 or VEGF as seeds are shown in Figure 7.8. In both glioma subnetworks, we used the influence as the strength of connection. The numbers over the arrows represent influences.

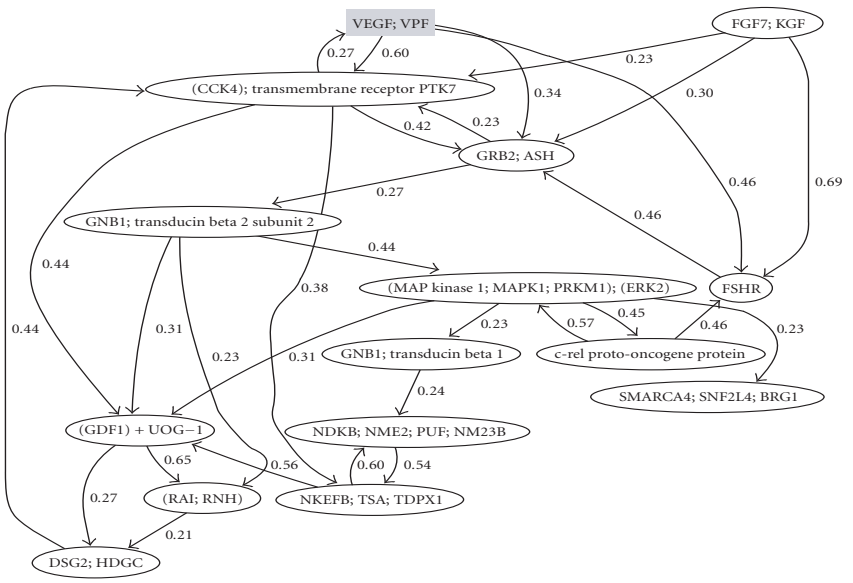
The subnetworks generated thus far have been constructed using the "seed growing" algorithm, starting from a large inferred network. All predictor functions in the starting network, consisting of almost 600 genes, were inferred using the CoD. A useful research aim is to repeat the inference using the best-fit extension method and reconstruct the subnetworks again. It is quite possible that the resultant subnetworks may exhibit some differences from those that have already been constructed. This possibility would furnish one with two opportunities. Firstly, it may reveal other genetic interactions that were not made apparent from the CoD-based inference method, in turn providing new hypotheses to be experimentally tested. Secondly, relationships consistent in both inference methods would strengthen one's confidence in the model-based results.

7.6.2. Interpretation and validation of subnetworks with prior biological knowledge and experimental results

Having constructed subnetworks in Figure 7.8 from expression via the seed-based growing algorithm, we would like to interpret and validate (in so far as that is



(a)



(b)

Figure 7.8. Two subnetworks generated from PBN modeling applied to a set of human glioma transcriptome data generated in our laboratory. (a) The subnetwork has been "grown" from the IGFBP2 (insulin-like growth factor binding protein 2) "seed." (b) The subnetwork has been grown from the VEGF (vascular endothelial growth factor) seed.

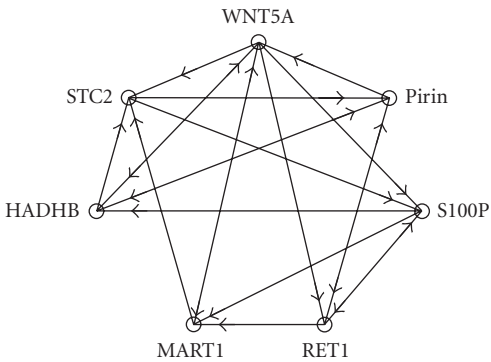


Figure 7.9. The seven-gene WNT5A network.

possible) the constructions using prior knowledge and independent experimental results. IGFBP2 and VEGF are genes that have been extensively studied and well characterized in the literature. It is known that IGFBP2 and VEGF are overexpressed in high-grade gliomas, glioblastoma multiforme (GBM)—the most advanced stage of tumor (Kleihues and Cavenee, WHO [99])—as compared to other types of glioma (Fuller et al. [100]). This finding was confirmed by two independent studies (Sallinen et al. [101], Elmlinger et al. [102]). Ongoing functional studies in the Cancer Genomics Laboratory (MD Anderson Cancer Center) using cell lines showed that when IGFBP2 is overexpressed, the cells become more invasive.

Studies that were completely independent of the PBN modeling work showed that NFκB activity is activated in cells stably overexpressing IGFBP2. This was done by using a luciferase reporter gene linked to a promoter element that contains an NFκB binding site. An analysis of the IGFBP2 promoter sequence showed that there are several NFκB binding sites, suggesting that NFκB transcriptionally regulates IGFBP2. A review of the literature revealed that Cazals et al. [103] indeed demonstrated that NFκB activated the IGFBP2-promoter in lung alveolar epithelial cells. Interestingly, in the IGFBP2 network (Figure 7.8a), we see an arrow linking NFκB3 to IGFBP2, and we see a two-step link from IGFBP2 to NFκB through TNF receptor 2 (TNFR2) and integrin-linked kinase (ILK). This parallels what was observed in the Cancer Genomics Laboratory. The presence of NFκB binding sites on the IGFBP2 promoter implies a direct influence of NFκB on IGFBP2. Although higher NFκB activity in IGFBP2 overexpressing cells was found, a transient transfection of IGFBP2 expressing vector together with NFκB promoter reporter gene construct did not lead to increased NFκB activity, suggesting an indirect effect of IGFBP2 on NFκB that will require time to take place. In fact, because of this indirect effect, this observation was not pursued for a year until the PBN-based subnetwork was linked to the laboratory experiments. IGFBP2 also contains an RGD domain, implying its interaction with integrin molecules. Integrin-linked kinase is in the integrin signal transduction pathway. The second subnetwork starting with VEGF (Figure 7.8b) offers even more compelling insight and supporting evidence.

Gliomas, like other cancers, are highly angiogenic, reflecting the need of cancer tissues for nutrients. To satisfy this need, expression of VEGF or vascular endothelial growth factor gene is often elevated. VEGF protein is secreted outside the cells and then binds to its receptor on the endothelial cells to promote their growth (Folkman [104]). Blockage of the VEGF pathway has been an intensive research area for cancer therapeutics (Bikfalvi and Bicknell [105]). A scrutiny of the VEGF network (Figure 7.8b) revealed some very interesting insight, which is highly consistent with prior biological knowledge derived from biochemical and molecular biology experiments. Let us elaborate. From the graph, VEGF, FGF7, FSHR, and PTK7 all influence Grb2. FGF7 is a member of fibroblast growth factor family (Rubin et al. [106]). FSHR is a follicle-stimulating hormone receptor. PTK7 is another tyrosine kinase receptor (Banga et al. [107]). The protein products of all four genes are part of signal transduction pathways that involve surface tyrosine kinase receptors. Those receptors, when activated, recruit a number of adaptor proteins to relay the signal to downstream molecules. Grb2 is one of the most crucial adaptors that have been identified (Stoletov et al. [108]). We should note that Grb2 is a target for cancer intervention (Wei et al. [109]) because of its link to multiple growth factor signal transduction pathways including VEGF, EGF, FGF, PDGF. Thus, the gene transcript relationships among the above five genes in the VEGF subnetwork appear to reflect their known or likely functional and physical relationship in cells. Molecular studies reported in the literature have further demonstrated that activation of protein tyrosine kinase receptor-Grb-2 complex in turn activates ras-MAP kinase- NFκB pathway to complete the signal relay from outside the cells to the nucleus of the cells (Bancroft et al. [110]). Although *ras* is not present on our VEGF network, a *ras* family member, GNB2, or transducing beta 2, is directly influenced by Grb2; GNB2 then influences MAP kinase 1 or ERK2, which in turn influences NFκB component c-rel (Pearson et al. [111]).

In the VEGF subnetwork shown in Figure 7.8, we also observe some potential feedback loop relationships. For example, c-rel influences FSHR, which influences Grb2-GNB2-MAPK1, and then influences c-rel itself. This may be a feedback regulation, a crucial feature of biological regulatory system in cells to maintain homeostasis. Other feedback regulation may also exist. RAI, or rel-A (another NFκB component) associated inhibitor (Yang et al. [112]), influences GNB2, which is two steps away from c-rel. RAI is further linked to PTK7 through GDF1, reflecting potentially another feedback regulatory mechanism. Whether those relationships are true negative feedback control mechanisms will need to be validated experimentally in the future. In this regard, the networks built from these models provide valuable theoretical guidance to experiments.

7.7. Perturbation and intervention

A major goal in gene network modeling is the ability to predict downstream effects on the gene network when a node is perturbed. This is very important for therapeutics. If we can predict the effect of such a perturbation, we can evaluate the virtue of a potential target when the effect on the entire system is considered.

The mathematical framework for performing this type of analysis has been developed by Shmulevich et al. [2]. Although this methodology has already been used for steady-state prediction, we have not attempted the prediction of downstream effects of specific perturbations. This, along with laboratory-based experimental verification, constitutes a valuable research goal.

A property of real gene regulatory networks is the existence of spontaneous emergence of ordered collective behavior of gene activity—that is, the evolution of networks into attractors. There is experimental evidence for the existence of attractors in real regulatory networks (Huang and Ingber [37]). As previously discussed, Boolean networks and PBNs also exhibit this behavior, the former with fixed-point and limit-cycle attractors (Kauffman [31]), the latter with absorbing states and irreducible sets of states (Shmulevich et al. [1, 53]). There is abundant justification in the assertion that in real cells, functional states, such as growth or quiescence, correspond to these attractors (Huang [30], Huang and Ingber [37]). Cancer is characterized by an imbalance between cellular states (attractors), such as proliferation and apoptosis (programmed cell death) resulting in loss of homeostasis.

As supported by Boolean network simulations, attractors are quite stable under most gene perturbations (Kauffman [31]). The same is true for real cellular states. However, a characteristic property of dynamical systems such as PBNs (and Boolean networks) is that the activity of some genes may have a profound effect on the global behavior of the entire system. That is to say, a change of value of *certain* genes at *certain* states of the network may drastically affect the values of many other genes in the long run and lead to different attractors. We should emphasize that the dependence on the current network state is crucial—a particular gene may exert a significant impact on the network behavior at one time, but that same gene may be totally ineffectual in altering the network behavior at a later time.

A detailed perturbation analysis, including the long-range effect of perturbations, has been carried out by Shmulevich et al. [2]. It was demonstrated that states of the network that are more "easily reachable" from other states (in terms of mean first-passage times) are more stable in the presence of gene perturbations. Consequently, these sets of states are those that correspond to cellular functional states and represent the probabilistic version of homeostatic stability of attractors in the PBN model.

Suppose, on the other hand, that we wish to elicit certain long-run behavior from the network. What genes would make the best candidates for intervention so as to increase the likelihood of this behavior? That is, suppose that the network is operating in a certain "undesirable" set of states and we wish to "persuade" it to transition into a "desirable" set of states by perturbing some gene. For practical reasons, we may wish to be able to intervene with as few genes as possible in order to achieve our goals. Such an approach can expedite the systematic search and identification of potential drug targets in cancer therapy.

This question was taken up by Shmulevich et al. in [2], where several methods for finding the best candidate genes for intervention, based on first-passage times, were developed. The first-passage times provide a natural way to capture the goals

of intervention in the sense that we wish to transition to certain states (or avoid certain states, if that is our goal) "as quickly as possible," or, alternatively, by maximizing the probability of reaching such states before a certain time. Suppose, for example, that we wish to persuade the network to flow into a set of states (irreducible subchain—the counterpart of an attractor) representing apoptosis (programmed cell death). This could be very useful, for example, in the case of cancer cells, which may keep proliferating. We may be able to achieve this action via the perturbation (intervention) of several different genes, but some of them may be better in the sense that the mean first-passage time to enter apoptosis is shorter.

The type of intervention described above—one that allows us to intervene with a gene—can be useful for modulating the dynamics of the network, but it is not able to alter the underlying structure of the network. Accordingly, the steady-state distribution remains unchanged. However, a lack of balance between certain sets of states, which is characteristic of neoplasia in view of gene regulatory networks, can be caused by mutations of the "wiring" of certain genes, thus permanently altering the state-transition structure and, consequently, the long-run behavior of the network (Huang [30]).

Therefore, it is prudent to develop a methodology for altering the steady-state probabilities of certain states or sets of states with minimal modifications to the rule-based structure. The motivation is that these states may represent different phenotypes or cellular functional states, such as cell invasion and quiescence, and we would like to decrease the probability that the whole network will end up in an undesirable set of states and increase the probability that it will end up in a desirable set of states. One mechanism by which we can accomplish this consists of altering some Boolean functions (predictors) in the PBN. For practical reasons, as above, we may wish to alter as few functions as possible. Such alterations to the rules of regulation may be possible by the introduction of a factor or drug that alters the extant behavior.

Shmulevich et al. [3] developed a methodology for altering the steady-state probabilities of certain states or sets of states, with minimal modifications to the underlying rule-based structure. This approach was framed as an optimization problem that can be solved using genetic algorithms, which are well suited for capturing the underlying structure of PBNs and are able to locate the optimal solution in a highly efficient manner. For example, in some computer simulations that were performed, the genetic algorithm was able to locate the optimal solution (structural alteration) in only 200 steps (evaluations of the fitness function), out of a total of 21 billion possibilities, which is the number of steps a brute-force approach would have to take. The reason for such high efficiency of the genetic algorithm is due to the embedded structure in the PBN that can be exploited.

7.8. External control

The aforementioned intervention methods do not provide effective "knobs" that could be used to externally guide the time evolution of the network towards more desirable states. By considering possible external interventions as control inputs,

and given a finite treatment horizon, ideas from optimal control theory can be applied to develop a general optimal intervention theory for Markovian gene regulatory networks, in particular, for PBNs. This strategy makes use of dynamic programming. The costs and benefits of using interventions are incorporated into a single performance index, which also penalizes the state where the network ends up following the intervention. The use of auxiliary variables makes sense from a biological perspective. For instance, in the case of diseases like cancer, auxiliary treatment inputs such as radiation, chemo-therapy, and so forth may be employed to move the state probability distribution vector away from one which is associated with uncontrolled cell proliferation or markedly reduced apoptosis. The auxiliary variables can include genes which serve as external master-regulators for all the genes in the network. To be consistent with the binary nature of the expression status of individual genes in the PBN, we will assume that the auxiliary variables (*control inputs*) can take on only the binary values zero or one. The values of the individual control inputs can be changed from one time step to the other in an effort to make the network behave in a desirable fashion. Interventions using full information (Datta et al. [7]) and partial information (Datta et al. [8]) have been considered for instantaneously random PBNs, for which the states of the Markov chain are the states of the PBN. Following Datta et al. [7], we summarize the full-information case here.

7.8.1. The optimal control problem

To develop the control strategy, let $\mathbf{x}(k) = [x_1(k), x_2(k), \ldots, x_n(k)]$ denote the state vector (gene activity profile) at step k for the n genes in the network. The state vector $\mathbf{x}(k)$ at any time step k is essentially an n-digit binary number whose decimal equivalent is given by

$$z(k) = 1 + \sum_{j=1}^{n} 2^{n-1} x_j(k). \tag{7.12}$$

As $\mathbf{x}(k)$ ranges from $000\cdots0$ to $111\cdots1$, $z(k)$ takes on all values from 1 to 2^n. The map from $\mathbf{x}(k)$ to $z(k)$ is one-to-one, onto, and hence invertible. Instead of the binary representation $\mathbf{x}(k)$ for the state vector, we can equivalently work with the decimal representation $z(k)$.

Suppose that the PBN has m control inputs, $u_1, u_2, \ldots, u_m$. Then at any given time step k, the row vector $\mathbf{u}(k) = [u_1(k), u_2(k), \ldots, u_m(k)]$ describes the complete status of all the control inputs. Clearly, $\mathbf{u}(k)$ can take on all binary values from $000\cdots0$ to $111\cdots1$. An equivalent decimal representation of the control input is given by

$$v(k) = 1 + \sum_{i=1}^{m} 2^{m-1} u_i(k). \tag{7.13}$$

As $\mathbf{u}(k)$ takes on binary values from $000\cdots0$ to $111\cdots1$, $\nu(k)$ takes on all values from 1 to 2^m. We can equivalently use $\nu(k)$ as an indicator of the complete control input status of the PBN at time step k.

As shown by Datta et al. [7], the one-step evolution of the probability distribution vector in the case of such a PBN with control inputs takes place according to the equation

$$\mathbf{w}(k+1) = \mathbf{w}(k)\mathbf{A}\big(\nu(k)\big), \tag{7.14}$$

where $\mathbf{w}(k)$ is the 2^n-dimensional state probability distribution vector and $\mathbf{A}(\nu(k))$ is the $2^n \times 2^n$ matrix of control-dependent transition probabilities. Since the transition probability matrix is a function of the control inputs, the evolution of the probability distribution vector of the PBN with control depends not only on the initial distribution vector but also on the values of the control inputs at different time steps.

In the control literature, (7.14) is referred to as a *controlled Markov chain* (Bertsekas [113]). Given a controlled Markov chain, the objective is to come up with a sequence of control inputs, usually referred to as a *control strategy*, such that an appropriate cost function is minimized over the entire class of allowable control strategies. To arrive at a meaningful solution, the cost function must capture the costs and benefits of using any control. The design of a good cost function is application dependent and likely to require considerable expert knowledge. In the case of diseases like cancer, treatment is typically applied over a finite time horizon. For instance, in the case of radiation treatment, the patient may be treated with radiation over a fixed interval of time, following which the treatment is suspended for some time as the effects are evaluated. After that, the treatment may be applied again but the important point to note is that the treatment window at each stage is usually finite. Thus we will be interested in a finite horizon problem where the control is applied only over a finite number of steps.

Suppose that the number of steps over which the control input is to be applied is M and we are interested in controlling the behavior of the PBN over the interval $k = 0, 1, 2, \ldots, M-1$. We can define a cost $C_k(z(k), \nu(k))$ as being the cost of applying the control input $\nu(k)$ when the state is $z(k)$. The expected cost of control over the entire treatment horizon is

$$E\left[\sum_{k=0}^{M-1} C_k\big(z(k), \nu(k)\big) \,|\, z(0)\right]. \tag{7.15}$$

Even if the network starts from a given (deterministic) initial state $z(0)$, the subsequent states will be random because of the stochastic nature of the evolution in (7.14). Consequently, the cost in (7.15) must be defined using an expectation. Expression (7.15) gives us one component of the finite horizon cost, namely the cost of control.

Regarding the second component of the cost, the net result of the control actions $v(0), v(1), \ldots, v(M-1)$ is that the state of the PBN will transition according to (7.14) and will end up in some state $z(M)$. Owing to the stochastic nature of the evolution, the terminal state $\mathbf{z}(M)$ is a random variable that can potentially take on any of the values $1, 2, \ldots, 2^n$. We assign a penalty, or terminal cost, $C_M(z(M))$ to each possible state. To do this, divide the states into different categories depending on their desirability and assign higher terminal costs to the undesirable states. For instance, a state associated with rapid cell proliferation leading to cancer should be associated with a high terminal penalty while a state associated with normal behavior should be assigned a low terminal penalty. For our purposes here, we will assume that the assignment of terminal penalties has been carried out and we have a terminal penalty $C_M(z(M))$ which is a function of the terminal state. This is the second component of our cost function. $C_M(z(M))$ is a random variable and so we must take its expectation while defining the cost function to be minimized. In view of (7.15), the finite horizon cost to be minimized is given by

$$E\left[\sum_{k=0}^{M-1} C_k(z(k), v(k)) + C_M(z(M)) \mid z(0)\right]. \tag{7.16}$$

To proceed further, let us assume that at time k, the control input $v(k)$ is a function of the current state $z(k)$, namely, $v(k) = \mu_k(z(k))$. The optimal control problem can now be stated: given an initial state $z(0)$, find a control law $\pi = [\mu_0, \mu_1, \ldots, \mu_{M-1}]$ that minimizes the cost functional

$$J_\pi(z(0)) = E\left[\sum_{k=0}^{M-1} C_k(z(k), \mu_k(z(k))) + C_M(z(M))\right] \tag{7.17}$$

subject to the probability constraint

$$P[z(k+1) = j \mid z(k) = i] = a_{ij}(v(k)), \tag{7.18}$$

where $a_{ij}(v(k))$ is the ith row, jth column entry of the matrix $\mathbf{A}(v(k))$. Optimal control problems of the type described by the preceding two equations can be solved by using *dynamic programming*, a technique pioneered by Bellman in the 1960s. We will not pursue the solution here, instead referring the reader to Datta et al. [7] for the complete solution. We will, however, follow Datta et al. [7] in providing an application.

7.8.2. Control of WNT5A in metastatic melanoma

In expression profiling studies concerning metastatic melanoma, the abundance of mRNA for the gene WNT5A was found to be a highly discriminating difference between cells with properties typically associated with high metastatic competence

versus those with low metastatic competence (Bittner et al. [114]; Weeraratna et al. [115]). In this study, experimentally increasing the levels of the WNT5A protein secreted by a melanoma cell line via genetic engineering methods directly altered the metastatic competence of that cell as measured by the standard in vitro assays for metastasis. A further finding of interest was that an intervention that blocked the WNT5A protein from activating its receptor, the use of an antibody that binds WNT5A protein, could substantially reduce WNT5A's ability to induce a metastatic phenotype. This suggests a study of control based on interventions that alter the contribution of the WNT5A gene's action to biological regulation, since the available data suggests that disruption of this influence could reduce the chance of a melanoma metastasizing, a desirable outcome.

The methods for choosing the 10 genes involved in a small local network that includes the activity of the WNT5A gene and the rules of interaction have been described by Kim et al. [9]. The expression status of each gene was quantized to one of three possible levels: -1 (down-regulated), 0 (unchanged), and 1 (up-regulated). Although the network is ternary valued instead of binary valued, the PBN formulation extends directly, with the terminology "probabilistic gene regulatory network" being applied instead of probabilistic Boolean network (Zhou et al. [10, 61]). The control theory also extends directly. Indeed, to apply the control algorithm of (Datta et al. [7]), it is not necessary to actually construct a PBN; all that is required are the transition probabilities between the different states under the different controls. For this study, the number of genes was reduced from 10 to 7 by using CoD analysis. The resulting genes along with their multivariate relationships are shown in Figure 7.9.

The control objective for this seven-gene network is to externally down-regulate the WNT5A gene. The reason is that it is biologically known that WNT5A ceasing to be down-regulated is strongly predictive of the onset of metastasis. For each gene in this network, its two best two-gene predictors were determined, along with their corresponding CoDs. Using the procedure by Shmulevich et al. in [1], the CoD information was used to determine the seven-by-seven matrix of transition probabilities for the Markov chain corresponding to the dynamic evolution of the seven-gene network.

The optimal control problem can now be completely specified by choosing (i) the treatment/intervention window, (ii) the terminal penalty, and (iii) the types of controls and the costs associated with them. For the treatment window, a window of length 5 was arbitrarily chosen, that is, control inputs would be applied only at time steps 0, 1, 2, 3, and 4. The terminal penalty at time step 5 was chosen as follows. Since the objective is to ensure that WNT5A is down regulated, a penalty of zero was assigned to all states for which WNT5A equals -1, a penalty of 3 to all states for which WNT5A equals 0, and a penalty of 6 to all states for which WNT5A equals 1. Here the choice of the numbers 3 and 6 is arbitrary but they do reflect our attempt to capture the intuitive notion that states where WNT5A equals 1 are less desirable than those where WNT5A equals 0. Two types of possible controls were considered by Datta et al. [7]; here only one of them was considered, where WNT5A is controlled via pirin.

The control objective is to keep WNT5A down-regulated. The control action consists of either forcing pirin to -1 or letting it remain wherever it is. A control cost of 1 is incurred if and only if pirin has to be forcibly reset to -1 at that time step. Using the resulting optimal controls, the evolution of the state probability distribution vectors has been studied with and without control. For every possible initial state, the resulting simulations have indicated that, at the final state, the probability of WNT5A being equal to -1 is higher with control than that without control; however, the probability of WNT5A being equal to -1 at the final time point is not, in general, equal to 1. This is not surprising given that one is trying to control the expression status of WNT5A using another gene and the control horizon of length 5 simply may not be adequate for achieving the desired objective with such a high probability. Nevertheless, even in this case, if the network starts from the state corresponding to STC2 $= -1$, HADHB $= 0$, MART-1 $= 0$, RET-1 $= 0$, S100P $= -1$, pirin $= 1$, WNT5A $= 1$ and evolves under optimal control, then the probability of WNT5A $= -1$ at the final time point equals 0.673521. This is quite good in view of the fact that the same probability would have been equal to 0 in the absence of any control action.

Bibliography

[1] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[2] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.

[3] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention," *Journal of Biological Systems*, vol. 10, no. 4, pp. 431–445, 2002.

[4] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proc. IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[5] E. R. Dougherty and I. Shmulevich, "Mappings between probabilistic Boolean networks," *Signal Processing*, vol. 83, no. 4, pp. 799–809, 2003.

[6] I. Shmulevich, I. Gluhovsky, R. Hashimoto, E. R. Dougherty, and W. Zhang, "Steady-state analysis of probabilistic Boolean networks," *Comparative and Functional Genomics*, vol. 4, no. 6, pp. 601–608, 2003.

[7] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, no. 1-2, pp. 169–191, 2003.

[8] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks: the imperfect information case," *Bioinformatics*, vol. 20, no. 6, pp. 924–930, 2004.

[9] S. Kim, H. Li, E. R. Dougherty, et al., "Can Markov chain models mimic biological regulation?," *Journal of Biological Systems*, vol. 10, no. 4, pp. 337–357, 2002.

[10] X. Zhou, X. Wang, and E. R. Dougherty, "Construction of genomic networks using mutual-information clustering and reversible-jump Markov-Chain-Monte-Carlo predictor design," *Signal Processing*, vol. 83, no. 4, pp. 745–761, 2003.

[11] R. F. Hashimoto, S. Kim, I. Shmulevich, W. Zhang, M. L. Bittner, and E. R. Dougherty, "Growing genetic regulatory networks from seed genes," *Bioinformatics*, vol. 20, no. 8, pp. 1241–1247, 2004.

[12] I. Shmulevich, "Model selection in genomics," *Environ. Health Perspect.*, vol. 111, no. 6, pp. A328–A329, 2003.

[13] E. P. van Someren, L. F. A. Wessels, and M. J. T. Reinders, "Linear modeling of genetic networks from experimental data," in *Proc. 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*, pp. 355–366, San Diego, Calif, USA, August 2000.

[14] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear modeling of mRNA expression levels during CNS development and injury," in *Pac. Symp. Biocomput. (PSB '99)*, vol. 4, pp. 41–52, Hawaii, USA, January 1999.

[15] K. Murphy and S. Mian, "Modelling gene expression data using dynamic Bayesian networks," Tech. Rep., University of California, Berkeley, Calif, USA, 1999.

[16] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian network to analyze expression data," *J. Computational Biology*, vol. 7, pp. 601–620, 2000.

[17] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young, "Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks," in *Pac. Symp. Biocomput. (PSB '01)*, pp. 422–433, Hawaii, USA, January 2001.

[18] E. J. Moler, D. C. Radisky, and I. S. Mian, "Integrating naive Bayes models and external knowledge to examine copper and iron homeostasis in S. cerevisiae," *Physiol. Genomics*, vol. 4, no. 2, pp. 127–135, 2000.

[19] D. C. Weaver, C. T. Workman, and G. D. Stormo, "Modeling regulatory networks with weight matrices," in *Pac. Symp. Biocomput. (PSB '99)*, vol. 4, pp. 112–123, Hawaii, USA, January 1999.

[20] T. Mestl, E. Plahte, and S. W. Omholt, "A mathematical framework for describing and analysing gene regulatory networks," *J. Theor. Biol.*, vol. 176, no. 2, pp. 291–300, 1995.

[21] T. Chen, H. L. He, and G. M. Church, "Modeling gene expression with differential equations," in *Pac. Symp. Biocomput. (PSB '99)*, vol. 4, pp. 29–40, Hawaii, USA, January 1999.

[22] J. Goutsias and S. Kim, "A nonlinear discrete dynamical model for transcriptional regulation: construction and properties," *Biophys. J.*, vol. 86, no. 4, pp. 1922–1945, 2004.

[23] H. H. McAdams and A. Arkin, "Stochastic mechanisms in gene expression," *Proc. Natl. Acad. Sci. USA*, vol. 94, no. 3, pp. 814–819, 1997.

[24] A. Arkin, J. Ross, and H. H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells," *Genetics*, vol. 149, no. 4, pp. 1633–1648, 1998.

[25] P. Smolen, D. A. Baxter, and J. H. Byrne, "Mathematical modeling of gene networks," *Neuron*, vol. 26, no. 3, pp. 567–580, 2000.

[26] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: in numero molecular biology," *Nat. Rev. Genet.*, vol. 2, no. 4, pp. 268–279, 2001.

[27] H. de Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *J. Comput. Biol.*, vol. 9, no. 1, pp. 69–103, 2002.

[28] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, 1969.

[29] L. Glass and S. A. Kauffman, "The logical analysis of continuous, non-linear biochemical control networks," *J. Theor. Biol.*, vol. 39, no. 1, pp. 103–129, 1973.

[30] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery," *J. Mol. Med.*, vol. 77, no. 6, pp. 469–480, 1999.

[31] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, NY, USA, 1993.

[32] R. Somogyi and C. Sniegoski, "Modeling the complexity of gene networks: understanding multigenic and pleiotropic regulation," *Complexity*, vol. 1, pp. 45–63, 1996.

[33] M. Aldana, S. Coppersmith, and L. P. Kadanoff, "Boolean dynamics with random couplings," in *Perspectives and Problems in Nonlinear Science*, E. Kaplan, J. E. Marsden, and K. R. Sreenivasan, Eds., Applied Mathematical Sciences Series, pp. 23–89, Springer-Verlag, New York, NY, USA, 2003.

[34] J. W. Bodnar, "Programming the Drosophila embryo," *J. Theor. Biol.*, vol. 188, no. 4, pp. 391–445, 1997.

[35] C. H. Yuh, H. Bolouri, and E. H. Davidson, "Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene," *Science*, vol. 279, no. 5358, pp. 1896–1902, 1998.

[36] L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla, "Genetic control of flower morphogenesis in Arabidopsis thaliana: a logical analysis," *Bioinformatics*, vol. 15, no. 7-8, pp. 593–606, 1999.

[37] S. Huang and D. E. Ingber, "Regulation of cell cycle and gene activity patterns by cell shape: evidence for attractors in real regulatory networks and the selective mode of cellular control," to appear in InterJournal Genetics, http://www.interjournal.org.

[38] G. Lahav, N. Rosenfeld, A. Sigal, et al., "Dynamics of the p53-Mdm2 feedback loop in individual cells," *Nat. Genet.*, vol. 36, no. 2, pp. 147–150, 2004.

[39] I. Shmulevich and W. Zhang, "Binary analysis and optimization-based normalization of gene expression data," *Bioinformatics*, vol. 18, no. 4, pp. 555–565, 2002.

[40] X. Zhou, X. Wang, and E. R. Dougherty, "Binarization of microarray data on the basis of a mixture model," *Mol. Cancer Ther.*, vol. 2, no. 7, pp. 679–684, 2003.

[41] W. Zhang, I. Shmulevich, and J. Astola, *Microarray Quality Control*, John Wiley & Sons, New York, NY, USA, 2004.

[42] Y. Chen, E. R. Dougherty, and M. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Biomedical Optics*, vol. 2, no. 4, pp. 364–374, 1997.

[43] M. K. Kerr, E. H. Leiter, L. Picard, and G. A. Churchill, "Sources of variation in microarray experiments," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 2002.

[44] H. H. McAdams and A. Arkin, "It's a noisy business! Genetic regulation at the nanomolar scale," *Trends Genet.*, vol. 15, no. 2, pp. 65–69, 1999.

[45] Z. Szallasi and S. Liang, "Modeling the normal and neoplastic cell cycle with "realistic Boolean genetic networks": their application for understanding carcinogenesis and assessing therapeutic strategies," in *Pac. Symp. Biocomput. (PSB '98)*, vol. 3, pp. 66–76, Hawaii, USA, January 1998.

[46] A. Wuensche, "Genomic regulation modeled as a network with basins of attraction," in *Pac. Symp. Biocomput. (PSB '98)*, vol. 3, pp. 89–102, Hawaii, USA, January 1998.

[47] R. Thomas, D. Thieffry, and M. Kaufman, "Dynamical behaviour of biological regulatory networks—I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state," *Bull. Math. Biol.*, vol. 57, no. 2, pp. 247–276, 1995.

[48] S. Liang, S. Fuhrman, and R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," in *Pac. Symp. Biocomput. (PSB '98)*, vol. 3, pp. 18–29, Hawaii, USA, January 1998.

[49] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions," in *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '98)*, pp. 695–702, San Francisco, Calif, USA, January 1998.

[50] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," *Pac. Symp. Biocomput.*, vol. 4, pp. 17–28, 1999.

[51] T. Akutsu, S. Miyano, and S. Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways," *Bioinformatics*, vol. 16, no. 8, pp. 727–734, 2000.

[52] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[53] I. Shmulevich, A. Saarinen, O. Yli-Harja, and J. Astola, "Inference of genetic regulatory networks via best-fit extensions," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 2002.

[54] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, "On learning gene regulatory networks under the Boolean network model," *Machine Learning*, vol. 52, no. 1-2, pp. 147–167, 2003.

[55] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.

[56] S. Kim, E. R. Dougherty, Y. Chen, et al., "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.

[57] I. Shmulevich and S. A. Kauffman, "Activities and sensitivities in Boolean network models," *Phys. Rev. Lett.*, vol. 93, no. 4, p. 048701, 2004.

[58] E. R. Dougherty, M. Bittner, Y. Chen, et al., "Nonlinear filters in genomic control," in *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP '99)*, Antalya, Turkey, June 1999.

[59] S. Kim, E. R. Dougherty, M. L. Bittner, et al., "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *J. Biomed. Opt.*, vol. 5, no. 4, pp. 411–424, 2000.

[60] X. Zhou, X. Wang, and E. R. Dougherty, "Gene prediction using multinomial probit regression with Bayesian gene selection," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 115–124, 2004.

[61] X. Zhou, X. Wang, R. Pal, I. Ivanov, M. L. Bittner, and E. R. Dougherty, "A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks," *Bioinformatics*, vol. 20, no. 17, pp. 2918–2927, 2004.

[62] E. B. Suh, E. R. Dougherty, S. Kim, et al., "Parallel computation and visualization tools for code-termination analysis of multivariate gene-expression relations," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 2002.

[63] I. Ivanov and E. R. Dougherty, "Reduction mappings between probabilistic Boolean networks," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 125–131, 2004.

[64] A. L. Gartel and A. L. Tyner, "Transcriptional regulation of the p21(WAF1/CIP1) gene," *Exp. Cell Res.*, vol. 246, no. 2, pp. 280–289, 1999.

[65] E. Schneider, M. Montenarh, and P. Wagner, "Regulation of CAK kinase activity by p53," *Oncogene*, vol. 17, no. 21, pp. 2733–2741, 1998.

[66] A. Rzhetsky, T. Koike, S. Kalachikov, et al., "A knowledge model for analysis and simulation of regulatory networks," *Bioinformatics*, vol. 16, no. 12, pp. 1120–1128, 2000.

[67] U. Braga-Neto, R. Hashimoto, E. R. Dougherty, D. V. Nguyen, and R. J. Carroll, "Is cross-validation better than resubstitution for ranking genes?," *Bioinformatics*, vol. 20, no. 2, pp. 253–258, 2004.

[68] M. Brun, E. R. Dougherty, and I. Shmulevich, "Attractors in probabilistic Boolean networks: steady-state probabilities and classification," to appear in Signal Process.

[69] M. K. Cowles and B. P. Carlin, "Markov Chain Monte Carlo convergence diagnostics: a comparative study," *Journal of the American Statistical Association*, vol. 91, pp. 883–904, 1996.

[70] A. E. Raftery and S. Lewis, "How many iterations in the Gibbs sampler?" in *Bayesian Statistics*, J. O. Berger, J. M. Bernardo, A. P. Dawid, and A. F. M. Smith, Eds., vol. 4, pp. 763–773, Oxford University Press, Oxford, UK, 1992.

[71] T. N. Sato, Y. Qin, C. A. Kozak, and K. L. Audus, "Tie-1 and tie-2 define another class of putative receptor tyrosine kinase genes expressed in early embryonic vascular system," *Proc. Natl. Acad. Sci. USA*, vol. 90, no. 20, pp. 9355–9358, 1993.

[72] S. Y. Cheng, H. J. Huang, M. Nagane, et al., "Suppression of glioblastoma angiogenicity and tumorigenicity by inhibition of endogenous expression of vascular endothelial growth factor," *Proc. Natl. Acad. Sci. USA*, vol. 93, no. 16, pp. 8502–8507, 1996.

[73] S. Hayashi, M. Yamamoto, Y. Ueno, et al., "Expression of nuclear factor-kappa B, tumor necrosis factor receptor type 1, and c-Myc in human astrocytomas," *Neurol. Med. Chir. (Tokyo)*, vol. 41, no. 4, pp. 187–195, 2001.

[74] M. Arsura, M. Wu, and G. E. Sonenshein, "TGF beta 1 inhibits NF-kappa B/Rel activity inducing apoptosis of B cells: transcriptional activation of I kappa B alpha," *Immunity*, vol. 5, no. 1, pp. 31–40, 1996.

[75] E. D. Strauch, J. Yamaguchi, B. L. Bass, and J. Y. Wang, "Bile salts regulate intestinal epithelial cell migration by nuclear factor-kappa B-induced expression of transforming growth factor-beta," *J. Am. Coll. Surg.*, vol. 197, no. 6, pp. 974–984, 2003.

[76] C. D. Johnson, Y. Balagurunathan, K. P. Lu, et al., "Genomic profiles and predictive biological networks in oxidant-induced atherogenesis," *Physiol. Genomics*, vol. 13, no. 3, pp. 263–275, 2003.

[77] I. Shmulevich, O. Yli-Harja, J. Astola, and A. Korshunov, "On the robustness of the class of stack filters," *IEEE Trans. Signal Processing*, vol. 50, no. 7, pp. 1640–1649, 2002.

[78] T. E. Ideker, V. Thorsson, and R. M. Karp, "Discovery of regulatory interactions through perturbation: inference and experimental design," in *Pac. Symp. Biocomput. (PSB '00)*, vol. 5, pp. 302–313, Hawaii, USA, January 2000.

[79] R. M. Karp, R. Stoughton, and K. Y. Yeung, "Algorithms for choosing differential gene expression experiments," in *Proc. 3rd Annual International Conference on Computational Molecular Biology (RECOMB '99)*, pp. 208–217, Lyon, France, 1999.

[80] Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, and Y. Eguchi, "Development of a system for the inference of large scale genetic networks," in *Pac. Symp. Biocomput. (PSB '01)*, vol. 6, pp. 446–458, Hawaii, USA, January 2001.

[81] K. Noda, A. Shinohara, M. Takeda, S. Matsumoto, S. Miyano, and S. Kuhara, "Finding genetic network from experiments by weighted network model," *Genome Inform.*, vol. 9, pp. 141–150, 1998.

[82] I. Shmulevich, M. Gabbouj, and J. Astola, "Complexity of the consistency problem for certain Post classes," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 31, no. 2, pp. 251–253, 2001.

[83] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bittner, P. S. Meltzer, and J. M. Trent, "Ratio statistics of gene expression levels and applications to microarray data analysis," *Bioinformatics*, vol. 18, no. 9, pp. 1207–1215, 2002.

[84] E. J. Coyle and J. H. Lin, "Stack filters and the mean absolute error criterion," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 8, pp. 1244–1254, 1988.

[85] E. J. Coyle, J. H. Lin, and M. Gabbouj, "Optimal stack filtering and the estimation and structural approaches to image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 2037–2066, 1989.

[86] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, "Optimal weighted median filtering under structural constraints," *IEEE Trans. on Signal Processing*, vol. 43, no. 3, pp. 591–604, 1995.

[87] E. R. Dougherty and R. P. Loce, "Precision of morphological-representation estimators for translation-invariant binary filters: increasing and nonincreasing," *Signal Processing*, vol. 40, no. 2-3, pp. 129–154, 1994.

[88] E. R. Dougherty and Y. Chen, "Optimal and adaptive design of logical granulometric filters," *Adv. Imaging Electron Phys.*, vol. 117, pp. 1–71, 2001.

[89] E. R. Dougherty and J. Barrera, "Pattern recognition theory in nonlinear signal processing," *J. Math. Imaging Vision*, vol. 16, no. 3, pp. 181–197, 2002.

[90] S. A. Kauffman, "The large scale structure and dynamics of gene control circuits: an ensemble approach," *J. Theor. Biol.*, vol. 44, no. 1, pp. 167–190, 1974.

[91] S. E. Harris, B. K. Sawhill, A. Wuensche, and S. Kauffman, "A model of transcriptional regulatory networks based on biases in the observed regulation rules," *Complexity*, vol. 7, no. 4, pp. 23–40, 2002.

[92] I. Shmulevich, H. Lähdesmäki, and K. Egiazarian, "Spectral methods for testing membership in certain Post classes and the class of forcing functions," *IEEE Signal Processing Lett.*, vol. 11, no. 2, pp. 289–292, 2004.

[93] I. Shmulevich, H. Lähdesmäki, E. R. Dougherty, J. Astola, and W. Zhang, "The role of certain Post classes in Boolean network models of genetic networks," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 19, pp. 10734–10739, 2003.

[94] E. Post, "Introduction to a general theory of elementary propositions," *Amer. J. Math.*, vol. 43, pp. 163–185, 1921.

[95] E. I. Nechiporuk, "On the complexity of circuits in some bases, containing non-trivial elements with zero weights," *Prob. Cybern.*, , no. 8, 1962.

[96] A. A. Muchnik and S. G. Gindikin, "On the completeness of systems of unreliable elements which realize functions of the algebra of logic," *Dokl. Akad. Nauk SSSR*, vol. 144, no. 5, 1962.

[97] M. E. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. Suppl 1, pp. 2566–2572, 2002.

[98] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[99] P. Kleihues and W. K. Cavenee, Eds., *World Health Organization Classification of Tumours: Pathology and Genetics of Tumours of Nervous System*, Oxford University Press, Oxford, UK, 2000.

[100] G. N. Fuller, C. H. Rhee, K. R. Hess, et al., "Reactivation of insulin-like growth factor binding protein 2 expression in glioblastoma multiforme: a revelation by parallel gene expression profiling," *Cancer Res.*, vol. 59, no. 17, pp. 4228–4232, 1999.

[101] S. L. Sallinen, P. K. Sallinen, H. K. Haapasalo, et al., "Identification of differentially expressed genes in human gliomas by DNA microarray and tissue chip techniques," *Cancer Res.*, vol. 60, no. 23, pp. 6617–6622, 2000.

[102] M. W. Elmlinger, M. H. Deininger, B. S. Schuett, et al., "In vivo expression of insulin-like growth factor-binding protein-2 in human gliomas increases with the tumor grade," *Endocrinology*, vol. 142, no. 4, pp. 1652–1658, 2001.

[103] V. Cazals, E. Nabeyrat, S. Corroyer, Y. de Keyzer, and A. Clement, "Role for NF-kappa B in mediating the effects of hyperoxia on IGF-binding protein 2 promoter activity in lung alveolar epithelial cells," *Biochim. Biophys. Acta.*, vol. 1448, no. 3, pp. 349–362, 1999.

[104] J. Folkman, "Angiogenesis in cancer, vascular, rheumatoid and other disease," *Nat. Med.*, vol. 1, no. 1, pp. 27–31, 1995.

[105] A. Bikfalvi and R. Bicknell, "Recent advances in angiogenesis, anti-angiogenesis and vascular targeting," *Trends Pharmacol. Sci.*, vol. 23, no. 12, pp. 576–582, 2002.

[106] J. S. Rubin, H. Osada, P. W. Finch, W. G. Taylor, S. Rudikoff, and S. A. Aaronson, "Purification and characterization of a newly identified growth factor specific for epithelial cells," *Proc. Natl. Acad. Sci. USA*, vol. 86, no. 3, pp. 802–806, 1989.

[107] S. S. Banga, H. L. Ozer, and S. T. Park, S. K. Lee, "Assignment of PTK7 encoding a receptor protein tyrosine kinase-like molecule to human chromosome 6p21.1-p12.2 by fluorescence in situ hybridization," *Cytogenet. Cell Genet.*, vol. 76, no. 1-2, pp. 43–44, 1997.

[108] K. V. Stoletov, K. E. Ratcliffe, and B. I. Terman, "Fibroblast growth factor receptor substrate 2 participates in vascular endothelial growth factor-induced signaling," *FASEB J.*, vol. 16, no. 10, pp. 1283–1285, 2002.

[109] C. Q. Wei, Y. Gao, K. Lee, et al., "Macrocyclization in the design of Grb2 SH2 domain-binding ligands exhibiting high potency in whole-cell systems," *J. Med. Chem.*, vol. 46, no. 2, pp. 244–254, 2003.

[110] C. C. Bancroft, Z. Chen, G. Dong, et al., "Coexpression of proangiogenic factors IL-8 and VEGF by human head and neck squamous cell carcinoma involves coactivation by MEK-MAPK and IKK-NF-kappaB signal pathways," *Clin. Cancer Res.*, vol. 7, no. 2, pp. 435–442, 2001.

[111] G. Pearson, J. M. English, M. A. White, and M. H. Cobb, "ERK5 and ERK2 cooperate to regulate NF-kappaB and cell transformation," *J. Biol. Chem.*, vol. 276, no. 11, pp. 7927–7931, 2001.

[112] J.-P. Yang, M. Hori, T. Sanda, and T. Okamoto, "Identification of a novel inhibitor of nuclear factor-kappa-B, RelA-associated inhibitor," *J. Biol. Chem.*, vol. 274, no. 22, pp. 15662–15670, 1999.

[113] D. P. Bertsekas, *Dynamic Programming and Stochastic Control*, Academic Press, Orlando, Fla, USA, 1976.

[114] M. Bittner, P. Meltzer, and Y. Chen, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.

[115] A. T. Weeraratna, Y. Jiang, G. Hostetter, et al., "Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma," *Cancer Cell*, vol. 1, no. 3, pp. 279–288, 2002.

[116] S. Agaian, J. Astola, and K. Egiazarian, *Binary Polynomial Transforms and Nonlinear Digital Filters*, Marcel Dekker, New York, NY, USA, 1995.

[117] O. Coudert, "Doing two-level logic minimization 100 times faster," in *Proc. 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '98)*, pp. 112–121, Society for Industrial and Applied Mathematics, San Francisco, Calif, USA, January 1995.

[118] S. Dedhar, B. Williams, and G. Hannigan, "Integrin-linked kinase (ILK): a regulator of integrin and growth-factor signalling," *Trends Cell Biol.*, vol. 9, no. 8, pp. 319–323, 1999.

[119] E. R. Dougherty and S. N. Attoor, "Design issues and comparison of methods for microarray-based classification," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 2002.

[120] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, vol. 57 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, New York, NY, USA, 1993.

[121] C. J. Geyer and E. A. Thompson, "Annealing Markov Chain Monte Carlo with applications to ancestral inference," *Journal of the American Statistical Asssociation*, vol. 90, pp. 909–920, 1995.

[122] G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, "Markovian analysis of large finite state machines," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 12, pp. 1479–1493, 1996.

[123] P. L. Hammer, A. Kogan, and U. G. Rothblum, "Evaluation, strength, and relevance of variables of Boolean functions," *SIAM Journal on Discrete Mathematics*, vol. 13, no. 3, pp. 302–312, 2000.

[124] W. K. Hastings, "Monte Carlo sampling methods using Markov Chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[125] S. Kim, E. R. Dougherty, I. Shmulevich, et al., "Identification of combination gene sets for glioma classification," *Mol. Cancer Ther.*, vol. 1, no. 13, pp. 1229–1236, 2002.

[126] L. M. Loew and J. C. Schaff, "The virtual cell: a software environment for computational cell biology," *Trends Biotechnol.*, vol. 19, no. 10, pp. 401–406, 2001.

[127] D. A. McAllester, "PAC-Bayesian stochastic model selection," *Machine Learning*, vol. 51, no. 1, pp. 5–21, 2003.

[128] P. Mendes, W. Sha, and K. Ye, "Artificial gene networks for objective comparison of analysis algorithms," *Bioinformatics*, vol. 19, no. Suppl 2, pp. II122–II129, 2003.

[129] C. Mircean, I. Tabus, J. Astola, et al., "Quantization and similarity measure selection for discrimination of lymphoma subtypes under k-nearest neighbor classification," in *Proc. SPIE Photonics West, Biomedical Optics*, San Jose, Calif, USA, January 2004.

[130] M. Mitchell, J. P. Crutchfield, and P. T. Hraber, "Evolving cellular automata to perform computations: mechanisms and impediments," *Physica D.*, vol. 75, no. 1–3, pp. 361–391, 1994.

[131] S. R. Neves and R. Iyengar, "Modeling of signaling networks," *Bioessays*, vol. 24, no. 12, pp. 1110–1117, 2002.

[132] S. N. Nikolopoulos and C. E. Turner, "Integrin-linked kinase (ILK) binding to paxillin LD1 motif regulates ILK localization to focal adhesions," *J. Biol. Chem.*, vol. 276, no. 26, pp. 23499–23505, 2001.

[133] B. Plateau and K. Atif, "Stochastic automata network of modeling parallel systems," *IEEE Trans. Software Eng.*, vol. 17, no. 10, pp. 1093–1108, 1991.

[134] J. S. Rosenthal, "Minorization conditions and convergence rates for Markov chain Monte Carlo," *Journal of the American Statistical Asssociation*, vol. 90, no. 430, pp. 558–566, 1995.

[135] I. Shmulevich, O. Yli-Harja, K. Egiazarian, and J. Astola, "Output distributions of recursive stack filters," *IEEE Signal Processing Lett.*, vol. 6, no. 7, pp. 175–178, 1999.

[136] P. T. Spellman, G. Sherlock, M. Q. Zhang, et al., "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization," *Mol. Biol. Cell.*, vol. 9, no. 12, pp. 3273–3297, 1998.

[137] I. Tabus and J. Astola, "On the use of MDL principle in gene expression prediction," *EURASIP J. Appl. Signal Process.*, vol. 4, no. 4, pp. 297–303, 2001.

[138] A. Wagner, "How to reconstruct a large genetic network from n gene perturbations in fewer than n^2 easy steps," *Bioinformatics*, vol. 17, no. 12, pp. 1183–1197, 2001.

[139] H. Wang, H. Wang, W. Shen, et al., "Insulin-like growth factor binding protein 2 enhances glioblastoma invasion by activating invasion-enhancing genes," *Cancer Res.*, vol. 63, no. 15, pp. 4315–4321, 2003.

[140] D. E. Zak, F. J. Doyle, G. E. Gonye, and J. S. Schwaber, "Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data," in *Proc. 2nd International Conference on Systems Biology (ICSB '01)*, pp. 231–238, Pasadena, Calif, USA, November 2001.

[141] D. E. Zak, R. K. Pearson, R. Vadigepalli, G. E. Gonye, J. S. Schwaber, and F. J. Doyle III, "Continuous-time identification of gene expression models," *OMICS*, vol. 7, no. 4, pp. 373–386, 2003.

Ilya Shmulevich: Cancer Genomics Laboratory, Department of Pathology, University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA

Email: is@ieee.org

Edward R. Dougherty: Department of Electrical Engineering, Texas A&M University, TX 77843-3128, USA; Cancer Genomics Laboratory, Department of Pathology, University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA

Email: e_dougherty@tamu.edu

8 Bayesian networks for genomic analysis

Paola Sebastiani, Maria M. Abad, and Marco F. Ramoni

Bayesian networks are emerging into the genomic arena as a general modeling tool able to unravel the cellular mechanism, to identify genotypes that confer susceptibility to disease, and to lead to diagnostic models. This chapter reviews the foundations of Bayesian networks and shows their application to the analysis of various types of genomic data, from genomic markers to gene expression data. The examples will highlight the potential of this methodology as well as the current limitations and we will describe new research directions that hold the promise to make Bayesian networks a fundamental tool for genome data analysis.

8.1. Introduction

One of the most striking characteristics of today's biomedical research practice is the availability of genomic-scale information. This situation has been created by the simultaneous but not unrelated development of "genome-wide" technologies, mostly rooted in the Human Genome Project: fast sequencing techniques, high-density genotype maps, DNA, and protein microarrays. Sequencing and genotyping techniques have evolved into powerful tools to identify genetic variations across individuals responsible for predispositions to some disease, response to therapies, and other observable characters known as phenotypes. Single-nucleotide polymorphisms (SNPs)—a single-base variation across the individuals of a population—are considered the most promising natural device to uncover the genetic basis of common diseases. By providing a high-resolution map of the genome, they allow researchers to associate variations in a particular genomic region to observable traits [1, 2]. Commercially available technology, such as the Affymetrix GeneChip Mapping 10 K Array and Assay Set (http://affymetrix.com), is able to simultaneously genotype 10 000 SNPs in an individual. Other technologies are able to interrogate the genomic structure of a cell on a genome-wide scale: CGH microarrays are able to provide genome-wide identification of chromosomal imbalances—such as deletions and amplifications—that are common rearrangements in most tumors [3]. These rearrangements identify different tumor types or stages and this technology allows us to dive into the mutagenic structure of tumor tissues.

Despite their differences—large scale genotyping interrogates the normal DNA of an individual, while CGH microarrays are specifically designed to study mutagenic tissues like tumors—these two technologies focus on the identification of structural genomic information, that is, information about the DNA sequence of a cell. The functional counterparts of these genomic platforms, on the other hand, are designed to quantify the expression of the genes encoded by the DNA of a cell, as the amount of RNA produced by each single gene. cDNA and oligonucleotide microarrays [4, 5, 6] enable investigators to simultaneously measure the expression of thousands of genes and hold the promise to cast new light onto the regulatory mechanisms of the genome [7]. The ability they offer to observe the genome in action has opened the possibility of profiling gene behaviors, studying interactions among genes, and discovering new classes of diseases on the basis of their genomic profile alone. The rising field of proteomics takes this study one step forward to proteins—the final product of gene expression [8]—and, using mass spectrometry technology, investigators can now measure in parallel the entire protein complement in a given cell, tissue, or organism [9].

All these technologies come to join today long-term cohort studies, like the Nurses' Health Study (http://www.channing.harvard.edu/nhs) and the Framingham Heart Study (http://www.framingham.com/heart) that have been collecting detailed "phenome-wide" information about hundreds of thousands individuals over several decades. Although the individual contribution of each technology has already been invaluable, the potential of their integration is even greater, but their ability to deliver on their promise of understanding the fundamental rules of life and diseases rests on our ability to integrate this genomic information with large-scale phenotypic data [1]. The integration of information about genotypes, RNA expression, proteins, and phenotypes into a coherent landscape will lead not only to the discovery of clinical phenomena not observable at each individual level but also to a better understanding of the coding and regulatory mechanisms underpinning the expression of genes [10].

The main challenge of this endeavor is the identification of a common formalism able to model this massive amount of data. Bayesian networks (also known as directed graphical models) are a knowledge representation formalism born at the confluence of artificial intelligence and statistics that offer a powerful framework to model these different data sources. Bayesian networks have already been applied, by us and others, to the analysis of different types of genomic data—from gene expression microarrays [11, 12, 13, 14, 15] to protein-protein interactions [16] and genotype data [17, 18]—and their modular nature makes them easily extensible to the task of modeling these different types of data. However, the application of Bayesian networks to genomics requires the methodological development of new statistical and computational capabilities able to capture the complexity of genomic information.

This chapter will first describe the current state of the art about learning Bayesian networks from data. We will show the potential benefit of Bayesian networks as a model and reasoning tool through several examples. The examples will also highlight the limitations of the current methodology and we will describe new

research directions that hold the promise to make Bayesian networks a fundamental tool for genomic data analysis.

8.2. Fundamentals of Bayesian networks

Bayesian networks are a representation formalism at the cutting edge of knowledge discovery and data mining [19, 20, 21]. In this section, we will review the formalism of Bayesian networks and the process of learning them from databases.

8.2.1. Representation and reasoning

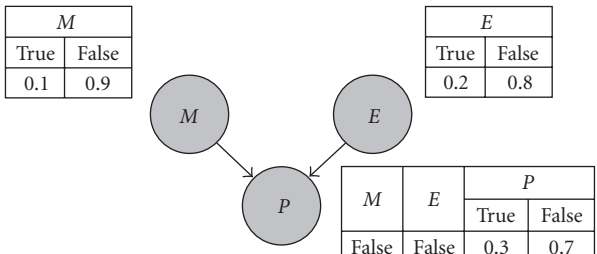
A Bayesian network has two components: a directed acyclic graph and a probability distribution. Nodes in the directed acyclic graph represent stochastic variables and arrows represent directed dependencies among variables that are quantified by conditional probability distributions.

As an example, consider the simple scenario in which a genetic marker together with an environmental condition create a phenotypic character. We describe the marker in the genetic code, the environmental condition, and the phenotypic character with three variables M, E, and P, each having two states "true" and "false." The Bayesian network in Figure 8.1 describes the dependency of the three variables with a directed acyclic graph, in which the two arrows pointing to the node P represent the joint action of the genetic marker and the environmental condition. Also, the absence of any directed arrow between the genetic marker and the environmental condition describes the *marginal independence* of the two variables that become dependent when we condition on the phenotype. Following the direction of the arrows, we call the node P a *child* of M and E, which become its *parents*. The Bayesian network in Figure 8.1 allows us to decompose the overall joint probability distribution of the three variables that would consist of $2^3 - 1 = 7$ parameters into three probability distributions, one conditional distribution for the variable P given the parents and two marginal distributions for the two parent variables M and E. These probabilities are specified by $1 + 1 + 4 = 6$ parameters. The decomposition is one of the key factors to provide both a verbal and a human understandable description of the system and to efficiently store and handle this distribution, which grows exponentially with the number of variables in the domain. The second key factor is the use of *conditional independence* between the network variables to break down their overall distribution into connected modules.

Suppose we have three random variables Y_1, Y_2, Y_3. Then Y_1 and Y_2 are independent given Y_3 if the conditional distribution of Y_1, given Y_2 and Y_3, is only a function of Y_3. Formally,

$$p(y_1|y_2, y_3) = p(y_1|y_3), \tag{8.1}$$

where $p(y|x)$ denotes the conditional probability/density of Y, given $X = x$. We use capital letters to denote random variables and small letters to denote their



M	
True	False
0.1	0.9

E	
True	False
0.2	0.8

M	E	P	
		True	False
False	False	0.3	0.7
False	True	0.6	0.4
True	False	0.7	0.3
True	True	0.9	0.1

Figure 8.1. A network describing the impact of a genetic marker (node M) and an environmental factor (node E) on a phenotypic character (node P). Each node in the network is associated with a probability table that describes the conditional distribution of the node, given its parents.

values. We also use the notation $Y_1 \perp Y_2 | Y_3$ to denote the conditional independence of Y_1 and Y_2 given Y_3.

Conditional and marginal independence are substantially different concepts. For example, two variables can be marginally independent, but they may be dependent when we condition on a third variable. The directed acyclic graph in Figure 8.1 shows this property: the two parent variables are marginally independent, but they become dependent when we condition on their common child. A well-known consequence of this fact is the Simpson's paradox [22] and a typical application in genetics is the dependency structure of genotypes among members of the same family: the genotypes of two parents are independent, assuming random mating, but they become dependent once the genotype of their common child is known.

Conversely, two variables that are marginally dependent may be made conditionally independent by introducing a third variable. This situation is represented by the directed acyclic graph in Figure 8.2, which shows two children nodes (Y_1 and Y_2) with a common parent Y_3. In this case, the two children nodes are independent, given the common parent, but they may become dependent when we marginalize the common parent out. Suppose, for example, the three variables represent the presence/absence of an X-linked genetic marker in the mother genotype (Y_3) and the children genotype (Y_1 and Y_2). The marginal distribution of Y_3 represents the prevalence of the marker in the population, and the conditional probabilities associated with the nodes Y_1 and Y_2 represent the probability that each child has the marker, given the maternal genotype. Then it is easy to compute the conditional probability that one of the two children has the marker, given that only the genotype of the other child is known. Because the probability of Y_2 changes according to the value of Y_1, the two variables are dependent. The seminal papers by Dawid [23, 24] summarize many important properties and alternative definitions of conditional independence.

The overall list of marginal and conditional independencies represented by the directed acyclic graph is summarized by the local and global Markov properties

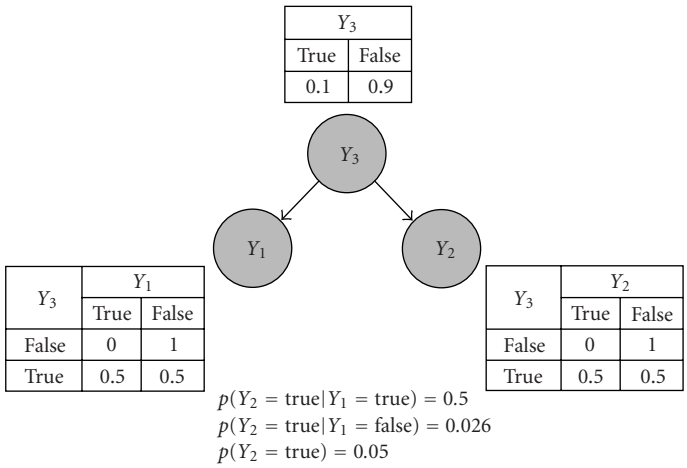


Figure 8.2. A network encoding the conditional independence of Y_1, Y_2 given the common parent Y_3. The panel in the middle shows that the distribution of Y_2 changes with Y_1 and hence the two variables are conditionally dependent.

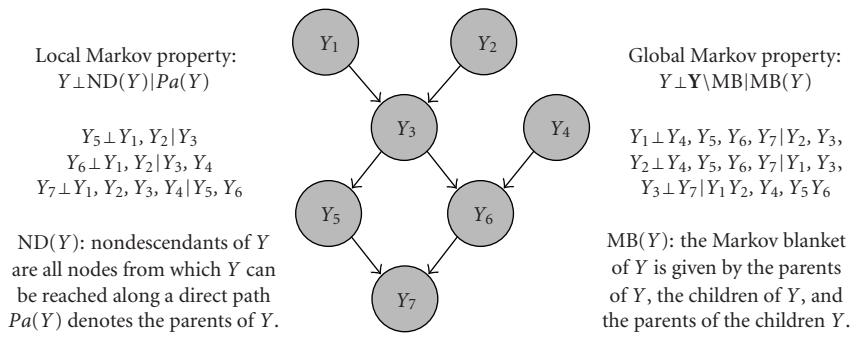


Figure 8.3. A Bayesian network with seven variables and some of the Markov properties represented by its directed acyclic graph. The panel on the left describes the local Markov property encoded by a directed acyclic graph and lists the three Markov properties that are represented by the graph in the middle. The panel on the right describes the global Markov property and lists three of the seven global Markov properties represented by the graph in the middle. The vector in bold denotes the set of variables represented by the nodes in the graph.

[25] that are exemplified in Figure 8.3 using a network of seven variables. The *local Markov property* states that each node is independent of its nondescendant given the parent nodes and leads to a direct factorization of the joint distribution of the network variables into the product of the conditional distribution of each variable Y_i given its parents Pa(y_i). Therefore, the joint probability (or density) of the v

network variables can be written as:

$$p(y_1,\ldots,y_v) = \prod_i p(y_i|\mathrm{pa}(y_i)). \tag{8.2}$$

In this equation, $\mathrm{pa}(y_i)$ denotes a set of values of $\mathrm{Pa}(Y_i)$. This property is the core of many search algorithms for learning Bayesian networks from data. With this decomposition, the overall distribution is broken into modules that can be interrelated, and the network summarizes all significant dependencies without information disintegration. Suppose, for example, the variables in the network in Figure 8.3 are all categorical. Then the joint probability $p(y_1,\ldots,y_7)$ can be written as the product of seven conditional distributions:

$$p(y_1)p(y_2)p(y_3|y_1,y_2)p(y_4)p(y_5|y_3)p(y_6|y_3,y_4)p(y_7|y_5,y_6). \tag{8.3}$$

The *global Markov property*, on the other hand, summarizes all conditional independencies embedded in the directed acyclic graph by identifying the Markov blanket of each node. This property is the foundation of many algorithms for probabilistic reasoning with Bayesian networks that allow the investigation of undirected relationships between the variables, and their use for making prediction and explanation. In the network in Figure 8.3, for example, we can compute the probability distribution of the variable Y_7, given that the variable Y_1 is observed to take a particular value (prediction) or, vice versa, we can compute the conditional distribution of Y_1 given the values of some other variables in the network (explanation). In this way, a Bayesian network becomes a complete simulation system able to forecast the value of unobserved variables under hypothetical conditions and, conversely, able to find the most probable set of initial conditions leading to the observed situation. Exact algorithms exist to perform this inference when the network variables are all discrete, all continuous, and modeled with Gaussian distributions, or when the network topology is constrained to particular structures [26, 27, 28].

For general network topologies and nonstandard distributions, we need to resort to stochastic simulation [29]. Among the several stochastic simulation methods currently available, Gibbs sampling [30, 31] is particularly appropriate for Bayesian network reasoning because of its ability to leverage on the graphical decomposition of joint multivariate distributions to improve computational efficiency. Gibbs sampling is also useful for probabilistic reasoning in Gaussian networks, as it avoids computations with joint multivariate distributions. Gibbs sampling is a Markov chain Monte Carlo method that generates a sample from the joint distribution of the nodes in the network. The procedure works by generating an ergodic Markov chain

$$\begin{pmatrix} y_{10} \\ \vdots \\ y_{v0} \end{pmatrix} \longrightarrow \begin{pmatrix} y_{11} \\ \vdots \\ y_{v1} \end{pmatrix} \longrightarrow \begin{pmatrix} y_{12} \\ \vdots \\ y_{v2} \end{pmatrix} \longrightarrow \cdots \tag{8.4}$$

that, under regularity conditions, converges to a stationary distribution. At each step of the chain, the algorithm generates y_{ik} from the conditional distribution of Y_i given all current values of the other nodes. To derive the marginal distribution of each node, the initial burn-in is removed, and the values simulated for each node are a sample generated from the marginal distribution. When one or more nodes in the network are observed, they are fixed in the simulation so that the sample for each node is from the conditional distribution of the node given the observed nodes in the network.

Gibbs sampling in directed graphical models exploits the Global Markov property, so that to simulate from the conditional distribution of one node Y_i given the current values of the other nodes, the algorithm needs to simulate from the conditional probability/density

$$p(y_i|y\backslash y_i) \propto p(y_i|\,\mathrm{pa}\,(y_i)) \prod_h p\Big(c(y_i)_h|\,\mathrm{pa}\,\big(c(y_i)_h\big)\Big), \tag{8.5}$$

where y denotes a set of values of all network variables, $\mathrm{pa}(y_i)$ and $c(y_i)$ are values of the parents and children of Y_i, $\mathrm{pa}(c(y_i)_h)$ are values of the parents of the hth child of Y_i, and the symbol "\" denotes the set difference.

8.2.2. Learning Bayesian networks from data

Learning a Bayesian network from data consists of the induction of its two different components: (1) the graphical structure of conditional dependencies (*model selection*); (2) the conditional distributions quantifying the dependency structure (*parameter estimation*). While the process of parameter estimation follows quite standard statistical techniques (see [32]), the automatic identification of the graphical model best fitting the data is a more challenging task. This automatic identification process requires two components: a scoring metric to select the best model and a search strategy to explore the space of possible, alternative models. This section will describe these two components—model selection and model search—and will also outline some methods to validate a graphical model once it has been induced from a data set.

8.2.2.1. Scoring metrics

We describe the traditional Bayesian approach to model selection that solves the problem as hypothesis testing. Other approaches based on independence tests or variants of the Bayesian metric like the minimum description length (MDL) score or the Bayesian information criterion (BIC) are described in [22, 25, 33]. We suppose to have a set $\mathcal{M} = \{M_0, M_1, \ldots, M_g\}$ of Bayesian networks, each network describing a hypothesis on the dependency structure of the random variables $Y_1, \ldots, Y_v$. Our task is to choose one network after observing a sample of data $\mathcal{D} = \{y_{1k}, \ldots, y_{vk}\}$, for $k = 1, \ldots, n$. By Bayes' theorem, the data $\mathcal{D}$ are used to revise the prior probability $p(M_h)$ of each model into the posterior probability,

which is calculated as

$$p(M_h|\mathcal{D}) \propto p(M_h)p(\mathcal{D}|M_h), \tag{8.6}$$

and the Bayesian solution consists of choosing the network with maximum posterior probability. The quantity $p(\mathcal{D}|M_h)$ is called the *marginal likelihood* and is computed by averaging out θ_h from the likelihood function $p(\mathcal{D}|\theta_h)$, where Θ_h is the vector parameterizing the distribution of $Y_1, \ldots, Y_v$, conditional on M_h. Note that, in a Bayesian setting, Θ_h is regarded as a random vector, with a prior density $p(\theta_h)$ that encodes any prior knowledge about the parameters of the model M_h. The likelihood function, on the other hand, encodes the knowledge about the mechanism underlying the data generation. In our framework, the data generation mechanism is represented by a network of dependencies and the parameters are usually a measure of the strength of these dependencies. By averaging out the parameters, the marginal likelihood provides an overall measure of the data generation mechanism that is independent of the values of the parameters. Formally, the marginal likelihood is the solution of the integral

$$p(\mathcal{D}|M_h) = \int p(\mathcal{D}|\theta_h)p(\theta_h)d\theta_h. \tag{8.7}$$

The computation of the marginal likelihood requires the specification of a parameterization of each model M_h that is used to compute the likelihood function $p(\mathcal{D}|\theta_h)$, and the elicitation of a prior distribution for Θ_h. The local Markov properties encoded by the network M_h imply that the joint density/probability of a case k in the data set can be written as

$$p(y_{1k}, \ldots, y_{vk}|\theta_h) = \prod_i p\left(y_{ik}|\,\mathrm{pa}\,(y_i)_k, \theta_h\right). \tag{8.8}$$

Here, $y_{1k}, \ldots, y_{vk}$ is the set of values (*configuration*) of the variables for the kth case, and $\mathrm{pa}(y_i)_k$ is the configuration of the parents of Y_i in case k. By assuming exchangeability of the data, that is, cases are independent given the model parameters, the overall likelihood is then given by the product

$$p(\mathcal{D}|\theta_h) = \prod_{ik} p\left(y_{ik}|\,\mathrm{pa}\,(y_i)_k, \theta_h\right). \tag{8.9}$$

Computational efficiency is gained by using priors for Θ_h that obey the directed hyper-Markov law [34, 35]. Under this assumption, the prior density $p(\theta_h)$ admits the same factorization of the likelihood function, namely, $p(\theta_h) = \prod_i p(\theta_{hi})$, where θ_{hi} is the subset of parameters used to describe the dependency of Y_i on its parents. This parallel factorization of the likelihood function and the prior density allows us to write

$$p(\mathcal{D}|M_h) = \prod_{ik} \int p\left(y_{ik}|\,\mathrm{pa}\,(y_i)_k, \theta_{hi}\right)p(\theta_{hi})d\theta_{hi} = \prod_i p(D|M_{hi}), \tag{8.10}$$

where $p(D|M_{hi}) = \prod_k \int p(y_{ik}|\,\mathrm{pa}(y_i)_k, \theta_{hi}) p(\theta_{hi}) d\theta_{hi}$. By further assuming decomposable network prior probabilities that factorize as $p(M_h) = \prod_i p(M_{hi})$ [36], the posterior probability of a model M_h is the product

$$p(M_h|\mathcal{D}) = \prod_i p(M_{hi}|\mathcal{D}). \tag{8.11}$$

Here $p(M_{hi}|\mathcal{D})$ is the posterior probability weighting the dependency of Y_i on the set of parents specified by the model M_h. Decomposable network prior probabilities are encoded by exploiting the modularity of a Bayesian network, and are based on the assumption that the prior probability of a local structure M_{hi} is independent of the other local dependencies M_{hj} for $j \neq i$. By setting $p(M_{hi}) = (g+1)^{-1/\nu}$, where $g+1$ is the cardinality of the model space and ν is the cardinality of the set of variables, there follows that uniform priors are also decomposable.

An important consequence of the likelihood modularity is that, in the comparison of models that differ for the parent structure of a variable Y_i, only the local marginal likelihood matters. Therefore, the comparison of two local network structures that specify different parents for the variable Y_i can be done by simply evaluating the product of the local *Bayes factor* $\mathrm{BF}_{hk} = p(\mathcal{D}|M_{hi})/p(\mathcal{D}|M_{ki})$, and the prior odds $p(M_h)/p(M_k)$, to compute the posterior odds of one model versus the other:

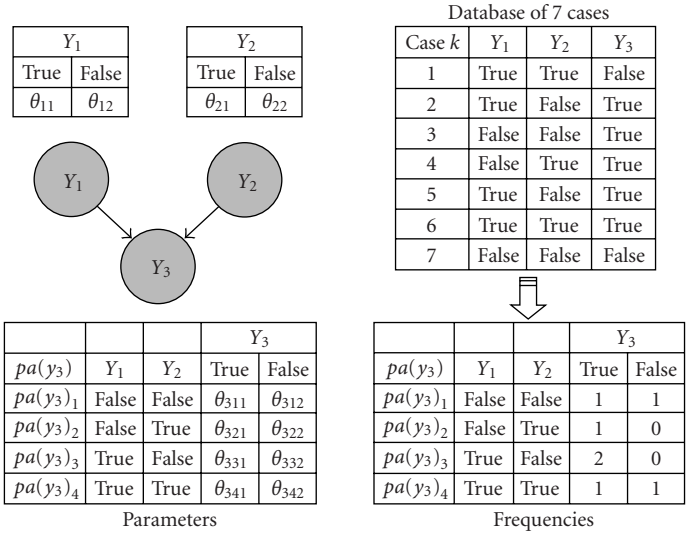
$$\frac{p(M_{hi}|\mathcal{D})}{p(M_{ki}|\mathcal{D})}. \tag{8.12}$$

The posterior odds provide an intuitive and widespread measure of fitness. Another important consequence of the likelihood modularity is that, when the models are a priori equally likely, we can learn a model locally by maximizing the marginal likelihood node by node.

When there are no missing data, the marginal likelihood $p(\mathcal{D}|M_h)$ can be calculated in closed form under the assumptions that all variables are discrete, or all variables follow Gaussian distributions and the dependencies between children and parents are linear. These two cases are described in the next examples. We conclude by noting that the calculation of the marginal likelihood of the data is the essential component for the calculation of the Bayesian estimate of the parameter θ_h, which is given by the expected value of the posterior distribution

$$p = (\theta_h|\mathcal{D}) = \frac{p(\mathcal{D}|\theta_h)p(\theta_h)}{p(\mathcal{D}|M_h)} = \prod_i \frac{p(\mathcal{D}|\theta_{hi})p(\theta_{hi})}{p(\mathcal{D}|M_{hi})}. \tag{8.13}$$

Example 1 (discrete variable networks). Suppose the variables $Y_1, \ldots, Y_\nu$ are all discrete, and denote by c_i the number of categories of Y_i. The dependency of each variable Y_i on its parents is represented by a set of *multinomial distributions* that describe the conditional distribution of Y_i on the configuration j of the parent



Y_1	
True	False
θ_{11}	θ_{12}

Y_2	
True	False
θ_{21}	θ_{22}

Case k	Y_1	Y_2	Y_3
1	True	True	False
2	True	False	True
3	False	False	True
4	False	True	True
5	True	False	True
6	True	True	True
7	False	False	False

			Y_3	
$pa(y_3)$	Y_1	Y_2	True	False
$pa(y_3)_1$	False	False	θ_{311}	θ_{312}
$pa(y_3)_2$	False	True	θ_{321}	θ_{322}
$pa(y_3)_3$	True	False	θ_{331}	θ_{332}
$pa(y_3)_4$	True	True	θ_{341}	θ_{342}

			Y_3	
$pa(y_3)$	Y_1	Y_2	True	False
$pa(y_3)_1$	False	False	1	1
$pa(y_3)_2$	False	True	1	0
$pa(y_3)_3$	True	False	2	0
$pa(y_3)_4$	True	True	1	1

Figure 8.4. A simple Bayesian network describing the dependency of Y_3 on Y_1 and Y_2 that are marginally independent. The table on the left describes the parameters θ_{3jk} ($j = 1,\ldots,4$ and $k = 1,2$) used to define the conditional distributions of $Y_3 = y_{3k}|\,\mathrm{pa}(y_3)_j$, assuming that all variables are binary. The two tables on the right describe a simple database of seven cases, and the frequencies n_{3jk}. The full joint distribution is defined by the parameters θ_{3jk} and the parameters θ_{1k} and θ_{2k} that specify the marginal distributions of Y_1 and Y_2.

variables $\mathrm{Pa}(Y_i)$. This representation leads to writing the likelihood function as

$$p(\mathcal{D}|\theta_h) = \prod_{ijk} \theta_{ijk}^{n_{ijk}}, \tag{8.14}$$

where the parameter θ_{ijk} denotes the conditional probability $p(y_{ik}|\,\mathrm{pa}(y_i)_j)$, n_{ijk} is the sample frequency of $(y_{ik}, \mathrm{pa}(y_i)_j)$, and $n_{ij} = \sum_k n_{ijk}$ is the marginal frequency of $\mathrm{pa}(y_i)_j$. Figure 8.4 shows an example of the notation for a network with three variables. With the data in this example, the likelihood function is written as

$$\{\theta_{11}^4\theta_{12}^3\}\{\theta_{21}^3\theta_{22}^4\}\{\theta_{311}^1\theta_{312}^1 \times \theta_{321}^1\theta_{322}^0 \times \theta_{331}^2\theta_{332}^0 \times \theta_{341}^1\theta_{342}^1\}. \tag{8.15}$$

The first two terms in the products are the contributions of nodes Y_1 and Y_2 to the likelihood, while the last product is the contribution of the node Y_3, with terms corresponding to the four conditional distributions of Y_3 given each of the four parent configurations.

The *hyper Dirichlet distribution* with parameters α_{ijk} is the conjugate hyper Markov law [34, 35] and it is defined by a density function proportional to the product $\prod_{ijk} \theta_{ijk}^{\alpha_{ijk}-1}$. This distribution encodes the assumption that the parameters θ_{ij} and $\theta_{i'j'}$ are independent for $i' \neq i$ and $j \neq j'$. These assumptions are known as *global and local parameter independence* [37], and are valid only under the assumption that the hyper-parameters α_{ijk} satisfy the consistency rule $\sum_j \alpha_{ij} = \alpha$ for all i

[38, 39]. Symmetric Dirichlet distributions satisfy easily this constraint by setting $\alpha_{ijk} = \alpha/(c_i q_i)$, where q_i is the number of states of the parents of Y_i. One advantage of adopting symmetric hyper Dirichlet priors in model selection is that, if we fix α constant for all models, then the comparison of posterior probabilities of different models is done conditionally on the same quantity α. With these parameterization and choice of prior distributions, the marginal likelihood is given by the equation

$$\prod_i p(\mathcal{D}|M_{hi}) = \prod_{ij} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_k \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{8.16}$$

where $\Gamma(\cdot)$ denotes the Gamma function, and the Bayesian estimate of the parameter θ_{ijk} is the posterior mean

$$E(\theta_{ijk}|\mathcal{D}) = \frac{\alpha_{ijk} + n_{ijk}}{\alpha_{ij} + n_{ij}}. \tag{8.17}$$

More details are in [32].

Example 2 (linear gaussian networks). Suppose now that the variables Y_1, $\ldots$, Y_v are all continuous, and the conditional distribution of each variable Y_i given its parents $\mathrm{Pa}(y_i) \equiv \{Y_{i1}, \ldots, Y_{ip(i)}\}$ follows a *Gaussian distribution* with a mean that is a linear function of the parent variables, and conditional variance $\sigma_i^2 = 1/\tau_i$. The parameter τ_i is called the precision. The dependency of each variable on its parents is represented by the linear regression equation

$$\mu_i = \beta_{i0} + \sum_j \beta_{ij} y_{ij} \tag{8.18}$$

that models the conditional mean of Y_i given the parent values y_{ij}. Note that the regression equation is additive (there are no interactions between the parent variables) to ensure that the model is graphical [25]. In this way, the dependency of Y_i on a parent Y_{ij} is equivalent to having the regression coefficient $\beta_{ij} \neq 0$. Given a set of exchangeable observations $\mathcal{D}$, the likelihood function is

$$p(\mathcal{D}|\theta_h) = \prod_i \left(\frac{\tau_i}{2\pi}\right)^{n/2} \prod_k \exp\left[-\frac{\tau_i (y_{ik} - \mu_{ik})^2}{2} \right], \tag{8.19}$$

where μ_{ik} denotes the value of the conditional mean of Y_i, in case k, and the vector θ_h denotes the set of parameters τ_i, β_{ij}. It is usually more convenient to use a matrix notation. We use the $n \times (p(i)+1)$ matrix X_i to denote the matrix of regression coefficients, with the kth row given by $(1, y_{i1k}, y_{i2k}, \ldots, y_{ip(i)k})$, β_i to denote the vector of parameters $(\beta_{i0}, \beta_{i1}, \ldots, \beta_{ip(i)})^T$ associated with Y_i, and, in this example, y_i to denote the vector of observations $(y_{i1}, \ldots, y_{in})^T$. With this notation, the likelihood can be written in a more compact form:

$$p(\mathcal{D}|\theta_h) = \prod_i \left(\frac{\tau_i}{2\pi}\right)^{n/2} \exp\left[-\frac{\tau_i (y_i - X_i\beta_i)^T (y_i - X_i\beta_i)}{2} \right]. \tag{8.20}$$

There are several choices to model the prior distribution on the parameters τ_i and β_i. For example, the conditional variance can be further parameterized as

$$\sigma_i^2 = V(Y_i) - \text{cov}(Y_i, \text{Pa}(y_i))\, V(\text{Pa}(y_i))^{-1}\, \text{cov}(\text{Pa}(y_i), Y_i), \tag{8.21}$$

where $V(Y_i)$ is the marginal variance of Y_i, $V(\text{Pa}(y_i))$ is the variance-covariance matrix of the parent variables, and $\text{cov}(Y_i, \text{Pa}(y_i))$ ($\text{cov}(\text{Pa}(y_i), Y_i)$) is the row (column) vector of covariances between Y_i and each parent Y_{ij}. With this parameterization, the prior on τ_i is usually a hyper-Wishart distribution for the joint variance-covariance matrix of Y_i, $\text{Pa}(y_i)$ [40]. The Wishart distribution is the multivariate generalization of a Gamma distribution. An alternative approach is to work directly with the conditional variance of Y_i. In this case, we estimate the conditional variances of each set of parents-child dependency and then the joint multivariate distribution that is needed for the reasoning algorithms is derived by multiplication. More details are described for example in [22, 41].

We focus on this second approach and again use the global parameter independence [37] to assign independent prior distributions to each set of parameters τ_i, β_i that quantify the dependency of the variable Y_i on its parents. In each set, we use the standard hierarchical prior distribution that consists of a marginal distribution for the precision parameter τ_i and a conditional distribution for the parameter vector β_i, given τ_i. The standard conjugate prior for τ_i is a *Gamma distribution*

$$\tau_i \sim \text{Gamma}(\alpha_{i1}, \alpha_{i2}), \qquad p(\tau_i) = \frac{1}{\alpha_{i2}^{\alpha_{i1}}\Gamma(\alpha_{i1})}\tau_i^{\alpha_{i1}-1}e^{-\tau_i/\alpha_{i2}}, \tag{8.22}$$

where

$$\alpha_{i1} = \frac{\nu_{io}}{2}, \qquad \alpha_{i2} = \frac{2}{\nu_{io}\sigma_{io}^2}. \tag{8.23}$$

This is the traditional Gamma prior for τ_i with hyper-parameters ν_{io} and σ_{io}^2 that can be given the following interpretation. The marginal expectation of τ_i is $E(\tau_i) = \alpha_{i1}\alpha_{i2} = 1/\sigma_{io}^2$ and

$$E\left(\frac{1}{\tau_i}\right) = \frac{1}{(\alpha_{i1}-1)\alpha_{i2}} = \frac{\nu_{io}\sigma_{io}^2}{\nu_{io}-2} \tag{8.24}$$

is the prior expectation of the population variance. Because the ratio $\nu_{io}\sigma_{io}^2/(\nu_{io}-2)$ is similar to the estimate of the variance in a sample of size ν_{io}, σ_{io}^2 is the prior population variance, based on ν_{io} cases seen in the past. Conditionally on τ_i, the prior density of the parameter vector β_i is supposed to be multivariate Gaussian:

$$\beta_i|\tau_i \sim N\left(\beta_{io}, (\tau_i R_{io})^{-1}\right), \tag{8.25}$$

where $\beta_{io} = E(\beta_i|\tau_i)$. The matrix $(\tau_i R_{io})^{-1}$ is the prior variance-covariance matrix of $\beta_i|\tau_i$ and R_{io} is the identity matrix so that the regression coefficients are a priori independent, conditionally on τ_i. The density function of β_i is

$$p(\beta_i|\tau_i) = \frac{\tau_i^{(p(i)+1)/2} \det(R_{io})^{1/2}}{(2\pi)^{(p(i)+1)/2}} e^{-\tau_i/2(\beta_i-\beta_{io})^T R_{io}(\beta_i-\beta_{io})}. \tag{8.26}$$

With this prior specifications, it can be shown that the marginal likelihood $p(\mathcal{D}|M_h)$ can be written in product form $\prod_i p(\mathcal{D}|M_{hi})$, where each factor is given by the quantity

$$p(\mathcal{D}|M_{hi}) = \frac{1}{(2\pi)^{n/2}} \frac{\det R_{io}^{1/2}}{\det R_{in}^{1/2}} \frac{\Gamma(\nu_{in}/2)}{\Gamma(\nu_{io}/2)} \frac{(\nu_{io}\sigma_{io}^2/2)^{\nu_{io}/2}}{(\nu_{in}\sigma_{in}^2/2)^{\nu_{in}/2}} \tag{8.27}$$

and the parameters are specified by the next updating rules:

$$\begin{aligned}
\alpha_{i1n} &= \frac{\nu_{io}}{2} + \frac{n}{2}, \\
\frac{1}{\alpha_{i2n}} &= \frac{-\beta_{in}^T R_{in}\beta_{in} + y_i^T y_i + \beta_{io}^T R_{io}\beta_{io}}{2} + \frac{1}{\alpha_{i2}}, \\
\nu_{in} &= \nu_{io} + n, \\
\sigma_{in} &= \frac{2}{\nu_{in}\alpha_{i2n}}, \\
R_{in} &= R_{io} + X_i^T X_i, \\
\beta_{in} &= R_{in}^{-1}(R_{io}\beta_{io} + X_i^T y_i).
\end{aligned} \tag{8.28}$$

The Bayesian estimates of the parameters are given by the posterior expectations

$$E(\tau_i|y_i) = \alpha_{i1n}\alpha_{i2n} = \frac{1}{\sigma_{in}^2}, \qquad E(\beta_i|y_i) = \beta_{in}, \tag{8.29}$$

and the estimate of σ_i^2 is $\nu_{in}\sigma_{in}^2/(\nu_{in}-2)$. More controversial is the use of improper prior distributions that describe lack of prior knowledge about the network parameters by uniform distributions [42]. In this case, we set $p(\beta_i, \tau_i) \propto \tau_i^{-c}$, so that $\nu_{io} = 2(1-c)$ and $\beta_{io} = 0$. The updated hyper-parameters are

$$\begin{aligned}
\nu_{in} &= \nu_{io} + n, \\
R_{in} &= X_i^T X_i, \\
\beta_{in} &= (X_i^T X_i)^{-1} X_i^T y_i \quad \text{(least squares estimate of } \beta\text{)}, \\
\sigma_{in} &= \frac{\mathrm{RSS}_i}{\nu_{in}}, \\
\mathrm{RSS}_i &= y_i^T y_i - y_i^T X_i (X_i^T X_i)^{-1} X_i^T y_i \quad \text{(residual sum of squares)},
\end{aligned} \tag{8.30}$$

and the marginal likelihood of each local dependency is

$$p(\mathcal{D}|M_{hi}) = \frac{1}{(2\pi)^{(n-p(i)-1)/2}}\Gamma\left(\frac{n-p(i)-2c+1}{2}\right)\left(\frac{\mathrm{RSS}_i}{2}\right)^{-(n-p(i)-2c+1)/2}\frac{1}{\det\left(X_i^T X_i\right)^{1/2}}. \tag{8.31}$$

A very special case is $c = 1$ that corresponds to $\nu_{io} = 0$. In this case, the local marginal likelihood simplifies to

$$p(\mathcal{D}|M_{hi}) = \frac{1}{(2\pi)^{(n-p(i)-1)/2}}\Gamma\left(\frac{n-p(i)-1}{2}\right)\left(\frac{\mathrm{RSS}_i}{2}\right)^{-(n-p(i)-1)/2}\frac{1}{\det\left(X_i^T X_i\right)^{1/2}}. \tag{8.32}$$

The estimates of the parameters σ_i and β_i become the traditional least squares estimates $\mathrm{RSS}_i/(\nu_{in}-2)$ and β_{in}. This approach can be extended to model an unknown variance-covariance structure of the regression parameters, using normal Wishart priors [41].

8.2.2.2. Model search

The likelihood modularity allows local model selection and simplifies the complexity of model search. Still, the space of the possible sets of parents for each variable grows exponentially with the number of candidate parents and successful heuristic search procedures (both deterministic and stochastic) have been proposed to render the task feasible [43, 44, 45, 46]. The aim of these heuristic search procedures is to impose some restrictions on the search space to capitalize on the decomposability of the posterior probability of each Bayesian network M_h. One suggestion, put forward in [43], is to restrict the model search to a subset of all possible networks that are consistent with an ordering relation $\succ$ on the variables $\{Y_1, \ldots, Y_v\}$. This ordering relation $\succ$ is defined by $Y_j \succ Y_i$ if Y_i cannot be parent of Y_j. In other words, rather than exploring networks with arrows having all possible directions, this order limits the search to a subset of networks in which there is only a subset of directed associations. At first glance, the requirement for an order among the variables could appear to be a serious restriction on the applicability of this search strategy, and indeed this approach has been criticized in the artificial intelligence community because it limits the automation of model search. From a modeling point of view, specifying this order is equivalent to specifying the hypotheses that need to be tested, and some careful screening of the variables in the data set may avoid the effort to explore a set of nonsensible models. For example, we have successfully applied this approach to model survey data [47, 48] and more recently genotype data [18]. Recent results have shown that restricting the search space by imposing an order among the variables yields a more regular space over the network structures [49].

In functional genomics, the determination of this order can be aided by the available information about gene control interactions embedded into known pathways. When the variables represent gene products, such as gene expression data, the order relationship can describe known regulatory mechanisms and it has been exploited for example in [14] to restrict the set of possible dependency structures between genes. This ordering operation can be largely automated by using some available programs, such as MAPPFinder [50] or GenMAPP [51], able to automatically map gene expression data to known pathways. For genes with unknown function, one can use different orders with random restarts. Other search strategies based on genetic algorithms [44], "ad hoc" stochastic methods [45], or Markov chain Monte Carlo methods [49] can also be used. An alternative approach to limit the search space is to define classes of equivalent directed graphical models [52].

The order imposed on the variables defines a set of candidate parents for each variable Y_i. One way to proceed is to implement an independent model selection for each variable Y_i and then link together the local models selected for each variable Y_i. A further reduction is obtained using the greedy search strategy deployed by the *K2 algorithm* [43]. The K2 algorithm is a bottom-up strategy that starts by evaluating the marginal likelihood of the model in which Y_i has no parents. The next step is to evaluate the marginal likelihood of each model with one parent only and if the maximum marginal likelihood of these models is larger than the marginal likelihood of the independence model, the parent that increases the likelihood most is accepted and the algorithm proceeds to evaluate models with two parents. If none of the models has marginal likelihood that exceeds that of the independence model, the search stops. The K2 algorithm is implemented in Bayesware Discoverer (http://www.bayesware.com) and the R-package deal [53]. Greedy search can be trapped in local maxima and it induces spurious dependency. A variant of this search to limit spurious dependency is stepwise regression [54]. However, there is evidence that the K2 algorithm performs as well as other search algorithms [55].

8.2.2.3. Validation

The automation of model selection is not without problems and both diagnostic and predictive tools are necessary to validate a multivariate dependency model extracted from data. There are two main approaches to model validation: one addresses the *goodness of fit* of the network selected from data and the other assesses the *predictive accuracy* of the network in some predictive/diagnostic tests.

The intuition underlying goodness-of-fit measures is to check the accuracy of the fitted model versus the data. In regression models in which there is only one dependent variable, the goodness of fit is typically based on some summary of the residuals that are defined by the difference between the observed data and the data reproduced by the fitted model. Because a Bayesian network describes a multivariate dependency model in which all nodes represent random variables, we developed *blanket residuals* [56] as follows. Given the network induced from data, for each case k in the database we compute the values fitted for each node Y_i,

given all the other values. Denote this fitted value by $\hat{y}_{ik}$ and note that, by the global Markov property, only the configuration in the Markov blanket of the node Y_i is used to compute the fitted value. For categorical variables, the fitted value $\hat{y}_{ik}$ is the most likely category of Y_i given the configuration of its Markov blanket, while for numerical variables the fitted value $\hat{y}_{ik}$ can be either the expected value of Y_i, given the Markov blanket, or the modal value. In both cases, the fitted values are computed by using one of the algorithms for probabilistic reasoning described in Section 8.2. By repeating this procedure for each case in the database, we compute fitted values for each variable Y_i, and then define the blanket residuals by

$$r_{ik} = y_{ik} - \hat{y}_{ik} \tag{8.33}$$

for numerical variables, and by

$$c_{ik} = \delta(y_{ik}, \hat{y}_{ik}) \tag{8.34}$$

for categorical variables, where the function $\delta(a, b)$ takes value $\delta = 0$ when $a = b$ and $\delta = 1$ when $a \neq b$. Lack of significant patterns in the residuals r_{ik} and approximate symmetry about 0 will provide evidence in favor of a good fit for the variable Y_i, while anomalies in the blanket residuals can help to identify weaknesses in the dependency structure that may be due to outliers or leverage points. Significance testing of the goodness of fit can be based on the standardized residuals

$$R_{ik} = \frac{r_{ik}}{\sqrt{V(y_i)}}, \tag{8.35}$$

where the variance $V(y_i)$ is computed from the fitted values. Under the hypothesis that the network fits the data well, we would expect to have approximately 95% of the standardized residuals within the limits $[-2, 2]$. When the variable Y_i is categorical, the residuals c_{ik} identify the error in reproducing the data and can be summarized to compute the error rate for fit.

Because these residuals measure the difference between the observed and fitted values, anomalies in the residuals can identify inadequate dependencies in the networks. However, residuals that are on average not significantly different from 0 do not necessarily prove that the model is good. A better validation of the network should be done on an independent test set to show that the model induced from one particular data set is *reproducible* and gives good predictions. Measures of the predictive accuracy can be the monitors based on the *logarithmic scoring function* [57]. The basic intuition is to measure the degree of surprise in predicting that the variable Y_i will take a value y_{ih} in the hth case of an independent test set. The measure of surprise is defined by the score

$$s_{ih} = -\log p\left(y_{ih} | \mathrm{MB}(y_i)_h\right), \tag{8.36}$$

where $\mathrm{MB}(y_i)_h$ is the configuration of the Markov blanket of Y_i in the test case h, $p(y_{ih} | \mathrm{MB}(y_i)_h)$ is the predictive probability computed with the model induced from data, and y_{ih} is the value of Y_i in the hth case of the test set. The score s_{ih} will be 0 when the model predicts y_{ih} with certainty, and increases as the probability of y_{ih} decreases. The scores can be summarized to derive *local and global monitors* and to define tests for predictive accuracy [40].

In the absence of an independent test set, standard cross-validation techniques are typically used to assess the predictive accuracy of one or more nodes [58]. In K-fold cross validation, the data are divided into K nonoverlapping sets of approximately the same size. Then $K - 1$ sets are used for retraining (or inducing) the network from data that is then tested on the remaining set using monitors or other measures of the predictive accuracy [59]. By repeating this process K times, we derive independent measures of the predictive accuracy of the network induced from data as well as measures of the robustness of the network to sampling variability. Note that the predictive accuracy based on cross validation is usually an overoptimistic measure, and several authors have recently argued that cross validation should be used with caution [60], particularly with small sample sizes.

8.3. Genomic applications of Bayesian networks

Bayesian networks have been applied to the analysis of several gene products, including gene expression measured with microarrays [11, 61] and proteins [16]. This section describes some applications of Bayesian networks in genomics. In the first two sections we use Bayesian networks to model the complex structure of gene-gene interactions in complex traits, using genetic markers and gene expression data measured with microarrays. The last section shows an application of Bayesian networks to proteomics. In all applications, the study design was a *case control* [62] with subjects selected according to their disease status: cases are subjects affected with the particular disease of interest, while controls are unaffected with the disease.

8.3.1. Networks of genetic markers

Many complex diseases are likely to be determined by the joint action of particular genotypes and their interaction with environmental factors. Alzheimer's disease is an example of a complex trait related to multiple genes and there is evidence that several genes and the environment influence the risk of this disease [62]. Another example is diabetes, for which several studies have identified different genotypes that are associated with the disease [63]. In both examples, polymorphic loci of several genes have been found to be associated with the disease.

It is well known that the majority of the DNA sequence is equal across all individuals except for a small proportion of positions that have more than one form (*allele*). A piece of DNA that has more than one form, each occurring with at least 1% frequency in the population, is called *polymorphic*, and when the piece is a

single base of the DNA, it is called a single nucleotide polymorphism (SNP). SNPs work as flags on a high-density map of the human genome and allow us to identify those genes whose polymorphisms may be causative of the disease [2]. In case-control studies, the data available for this discovery process are typically genotypes of case and control subjects at polymorphic loci, together with information about several clinical covariates and environmental factors. The genotype can be coded either as the presence/absence of the minor allele (the allele with smaller frequency in the population) in the two loci of the chromosome pair or as the complete allele pair that can be homozygous for the major allele, homozygous for the minor allele, or heterozygous when the two alleles are different.

The discovery of complex gene-environment interactions that confer susceptibility to disease requires advanced multivariate modeling tools. A typical solution is to resort to logistic regression models to describe the odds for the disease given a particular genotype. The advantages of logistic regression models are that they can be used to assess whether the association between the risk for disease and a particular genotype is *confounded* by some external factor (such as population admixture [64]) and they can be used to test whether an external factor or a particular genotype is an *effect modifier* of an association [65]. However, logistic regression models pose three serious limitations: when the susceptibility to disease is caused by the interaction among several genes, the number of parameters required to fit a logistic regression model increases at an exponential rate; the genotypes are treated as covariates rather than random variables; logistic regression is limited to examining the association between one phenotypic character at a time. To simultaneously overcome these three limitations, we have recently proposed to use Bayesian networks to discover the genetic makeup that confers susceptibility to overt stroke in patients with sickle cell anemia.

The complications of sickle cell anemia are likely to be determined by the actions of genes that modify the pathophysiology initiated by sickle hemoglobin. Overt stroke (CVA) occurs in about 10% of patients with sickle cell anemia. To define the genetic basis of CVA in sickle cell anemia, we examined the association of SNPs in several candidate genes of different functional classes with the likelihood of CVA. In our study, we considered 92 patients with a confirmed history of or incident complete nonhemorrhagic CVA, documented by imaging studies and 453 controls (patients who did not have a stroke in five years follow up). We modeled the genetic markers and their association with the CVA phenotype by Bayesian networks using the structural learning approach described in Section 8.2.2. We validated the network of association induced from data using cross validation, which showed that the network of gene-gene-phenotype interaction can predict the likelihood of CVA in patients with sickle cell anemia with 99.7% accuracy. We also validated the model using an independent set of 114 individuals with an accuracy of 98%. In both tests, the accuracy was measured by the frequency of individuals for whom the Bayesian network model predicted the correct phenotype with probability above 0.5 [66]. With this approach, we discovered a network of interacting genes that may confer susceptibility to CVA in patients with sickle cell anemia. Part of the network is displayed in Figure 8.5 and identifies polymorphisms of the

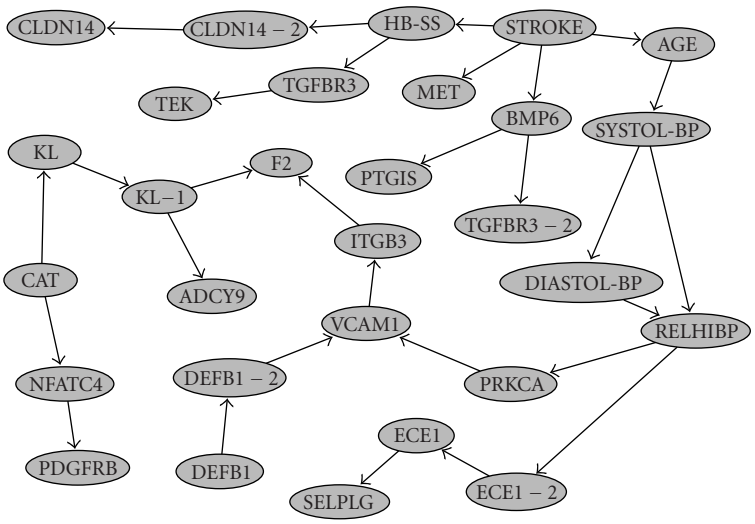


Figure 8.5. A Bayesian network representing a complex trait given by the interaction of several genes and clinical covariates.

genes *MET* and *BMP6* as directly associated with CVA. The Markov blanket of the node representing the phenotype (stroke) identifies the gene-gene-environment interaction that confers susceptibility to the disease. It consists of polymorphisms of the genes *MET* and *BMP6*, the age of the patient, and whether or not the patient is affected by α-thalassemia (node HB-SS). Dependencies between polymorphisms of other genes may be interpreted as an effect of population admixture, while dependencies between polymorphism of the same gene denote linkage disequilibrium [64].

8.3.2. Gene expression networks

The coherent probabilistic framework of Bayesian networks can be used not only to model genotype data but also gene expression data. Compared to standard expression profiling methods, Bayesian networks are able to represent the directionality of the influence among gene expression and they have already been deployed to understand both gene expression [12] and protein-protein interactions [16].

Another area of application of Bayesian networks in functional genomics is modeling differential expression in comparative experiments. Typical statistical techniques used to identify genes that have differential expression in two or more conditions work assuming that genes act independently [6]. Bayesian networks can be used to identify genes with differential expression by simultaneously modeling the structure of gene-gene interaction. Figure 8.6 provides an example that describes a network of gene expression interaction learned from a case-control study of prostate cancer. We used a data set of expression profiles derived from

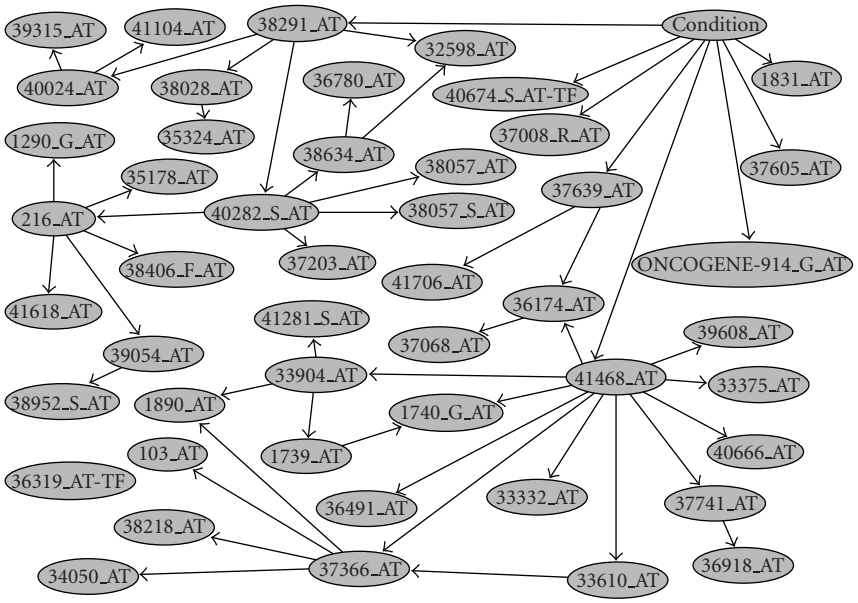


Figure 8.6. A Bayesian network associating genes that are differentially expressed between normal and tumor specimens (node condition). Genes are labelled by the Affymetrix probe ID.

102 prostatectomy specimens. Cases were 52 cancer specimens of patients undergoing surgery between 1996 and 1997, and controls were 50 normal specimens. The expression profiles were derived with the U95Av2 Affymetrix microarray and are described in [67]. We first analyzed the data with BADGE [68], a program for differential analysis that uses Bayesian model averaging to compute the posterior probability of differential expression. We selected about 200 genes with very large probability of differential expression and then modeled the network of interaction of gene expression. We used information about known functions of some genes to limit the search space and, for example, imposed the restriction that genes known as transcription factors could only be tested as parents of all other nodes. In the absence of an independent set, the final network was tested with 5-fold cross validation and had 93% accuracy in predicting the clinical status, and an average accuracy of about 80% in predicting the expression of each gene given the others.

Besides the identification of a molecular profile based on those genes that are directly related to the clinical status, the network displays some interesting associations. For example, changes in expression of *TRCγ* (41468_AT: a known enhancer of transcriptional activity specific for prostatic adenocarcinoma cell line) are associated with changes of expression of several genes including *SIM2* (39608_AT: a transcription repression), *PSMA* (1740_G_AT: a gene associated with prostate cancer), and *MRP* (36174_AT: a gene known as potential predictor of chemotherapy response). The probability of changes in expression of *Hepsin* (37639_AT: a gene with cell growth function) depends on both the clinical status and changes

in expression of *MRP*. The differential expression of the *Hepsin* gene influences changes in expression of *AMACR* (41706_AT: a marker of tumor differentiation known to be essential for growth of prostate cancer [69, 70].) These directed associations suggest a mechanism by which changes in the transcription factor TRCγ influence changes in genes involved in tumor growth. Another interesting fact is the directed association between *Adipsin* (40282_S_AT: a gene supposed to have a role in immune system biology) and *CRBP1* (38634_AT: a gene known to contribute to cancer by disrupting the vitamin A metabolism). One theory is that cancer arises from the accumulation of genetic changes that induce unlimited, self-sufficient growth and resistance to normal regulatory mechanisms, and these two sets of dependencies are consistent with this conjecture.

Of course, the nature of the data collected in a case-control study limits the dependency structure to represent associations rather than causal effects. This limitation is due to the data rather than the modeling approach, and data produced by controlled experiments have been used to induce causal networks [61, 71]. We will discuss this issue further in Section 8.4.3.

8.3.3. In silico integrative genomics

The predictive capabilities of Bayesian networks can be deployed for in silico identification of unobserved characteristics of the genome. Genetic studies are designed to identify regions of the genome associated with a disease phenotype. The success rate of these studies could be improved if we were able to predict in advance, before conducting the study, the likelihood of an SNP or a mutation in a particular region to be indeed pathogenic. To do so, we need to integrate the available information about SNPs and mutations with the available information about proteins, and predict that a particular change in the DNA will actually lead to a change in the encoded protein. Using Bayesian networks, we have developed a novel algorithm to predict pathogenic single amino acid changes, either nonsynonymous SNPs (nsSNPs)—SNPs causing a change in the encoded amino acid—or missense mutations, in conserved protein domains [17]. We found that the probability of a microbial missense mutation causing a change in phenotype depended on how much difference it made in several phylogenetic, biochemical, and structural features related to the single amino acid substitution. We tested our model on pathogenic allelic variants (missense mutations or nsSNPs) included in OMIM (www.ncbi.nlm.nih.gov/omim) and on the other nsSNPs in the same genes from dbSNP (www.ncbi.nlm.nih.gov/SNP) as the nonpathogenic variants. Our results show that our model was able to predict pathogenic variants with a 10% false-positive rate.

8.4. Advanced topics

This section describes some extensions of Bayesian networks to classification and for modeling nonlinear and temporal dependencies.

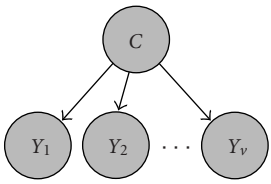


Figure 8.7. The structure of the naïve Bayes classifier.

8.4.1. Bayesian networks and classification

The goal of many studies in genomics medicine is the discovery of a molecular profile for disease diagnosis or prognosis. The molecular profile is typically based on gene expression [72, 73, 74]. Bayesian networks have been used in the past few years as supervised classification models able to discover and represent molecular profiles that characterize a disease [75, 76]. This section describes particular classification models that are simple Bayesian networks.

8.4.1.1. Classification

The term "supervised classification" covers two complementary tasks: the first is to identify a function mapping a set of *attributes* onto a *class*, and the other is to assign a class label to a set of unclassified cases described by attribute values. We denote by C the variable whose states represent the class labels c_i, and by Y_i the attributes. In our context, the class variable may represent a clinical status, and the attributes can be gene products such as gene expression data or genotypes.

Classification is typically performed by first training a classifier on a set of labelled cases (*training set*) and then using it to label unclassified cases (*test set*). The supervisory component of this classifier resides in the training signal, which provides the classifier with a way to assess a dependency measure between attributes and classes. The classification of a case with attribute values $y_{1k}, \ldots, y_{vk}$ is then performed by computing the probability distribution $p(C|y_{1k}, \ldots, y_{vk})$ of the class variable, given the attribute values, and by labelling the case with the most probable label. Most of the algorithms for learning classifiers described as Bayesian networks impose a restriction on the network structure, namely, that there cannot be arrows pointing to the class variable. In this case, by the local Markov property, the joint probability $p(y_{1k}, \ldots, y_{vk}, c_k)$ of class and attributes is factorized as $p(c_k)p(y_{1k}, \ldots, y_{vk}|c_k)$. The simplest example is known as a *naïve Bayes* (NB) classifier [77, 78] and makes the further simplification that the attributes Y_i are conditionally independent given the class C so that

$$p(y_{1k}, \ldots, y_{vk}|c_k) = \prod_i p(y_{ik}|c_k). \tag{8.37}$$

Figure 8.7 depicts the directed acyclic graph of an NB classifier. Because of the restriction on the network topology, the training step for an NB classifier consists

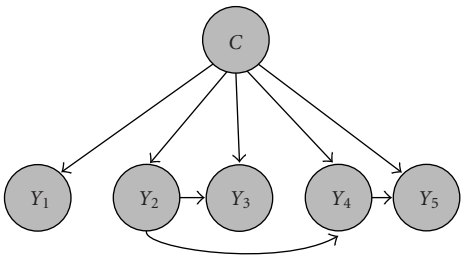


Figure 8.8. The structure of a TAN classifier.

of estimating the conditional probability distributions of each attribute, given the class, from a training data set. When the attributes are discrete or continuous variables and follow Gaussian distributions, the parameters are learned by using the procedure described in Section 8.2.2. Once trained, the NB classifies a case by computing the posterior probability distribution over the classes via Bayes' theorem and assigns the case to the class with the highest posterior probability.

Other classifiers have been proposed to relax the assumption that attributes are conditionally independent given the class. Perhaps the most competitive one is the *tree augmented naïve Bayes* (TAN) classifier [79] in which all the attributes have the class variable as a parent as well as another attribute. To avoid cycles, the attributes have to be ordered and the first attribute does not have other parents beside the class variable. Figure 8.8 shows an example of a TAN classifier with five attributes. An algorithm to infer a TAN classifier needs to choose both the dependency structure between attributes and the parameters that quantify this dependency. Due to the simplicity of its structure, the identification of a TAN classifier does not require any search but rather the construction of a tree among the attributes. An "ad hoc" algorithm called *construct-TAN* (CTAN) was proposed in [79]. One limitation of the CTAN algorithm to build TAN classifiers is that it applies only to discrete attributes, and continuous attributes need to be discretized.

Other extensions of the NB try to relax some of the assumptions made by the NB or the TAN classifiers. Some examples are the *l-limited dependence Bayesian classifier* (*l*-LDB) in which the maximum number of parents that an attribute can have is l [80]. Another example is the *unrestricted augmented naïve Bayes classifier* (ANB) in which the number of parents is unlimited but the scoring metric used for learning, the minimum description length criterion, biases the search toward models with small number of parents per attribute [79]. Due to the high dimensionality of the space of different ANB networks, algorithms that build this type of classifiers must rely on heuristic searches. More examples are reported in [79].

8.4.1.2. Molecular classification

Many learning algorithms show a high sensitivity to correlated features. In the case of data sets of gene expression profiles measured with microarrays, the large

Table 8.1. Test accuracies for some classifiers without and with feature selection.

Algorithm	All attributes	Feature selection
NB	75.4902	87.2549
sCTAN	73.5294	80.3922

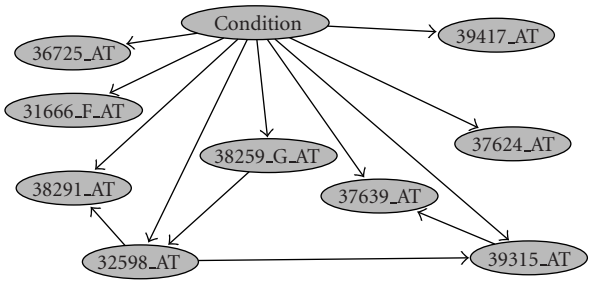


Figure 8.9. The structure of the TAN classifier with feature selection in the gene expression dataset.

number of genes must be drastically reduced in order to improve the diagnostic accuracy. Many learning algorithms that build classifiers and perform feature selection have been used in this context [81, 82, 83]. As an example, we used the NB and TAN classifiers to build a molecular classification model using the data set of gene expression measured in prostatectomy specimens (see Figure 8.6). Table 8.1, column 2, shows the test accuracy of the classifiers learned by different algorithms that was measured with 5-fold cross validation. The first classifier is an NB and the second classifier is a TAN. In both cases the parameters were learned with the Bayesian approach discussed in Section 8.2.2. Due the large number of input attributes, we used a filtered version of the wrapped feature selection algorithm described in [84] to increase the predictive accuracy.

Column 3 shows the accuracy of the same classifiers that were built by selecting a subset of the genes and shows that accuracy sensibly increases when feature selection is performed. The genes selected by the feature selection algorithm are 32598_AT, 38291_AT, 39315_AT, 37624_AT, 38059_G_AT, 36725_AT, 31666_F_AT, 39417_AT, 37639_AT and represent a molecular profile for classifying prostatectomy specimens into normal or tumor. Figure 8.9 shows the TAN structure chosen by the CTAN algorithm with feature selection. It is interesting to note that the selection of genes by the wrapped feature selection differs from that induced by the standard Bayesian algorithm described in Section 8.2.2. Particularly, neither of the classifiers reaches the classification accuracy of the Bayesian network model in Figure 8.6.

8.4.2. Generalized Gamma networks

Most of the work on learning Bayesian networks from data has focused on learning networks of categorical variables, or networks of continuous variables that are

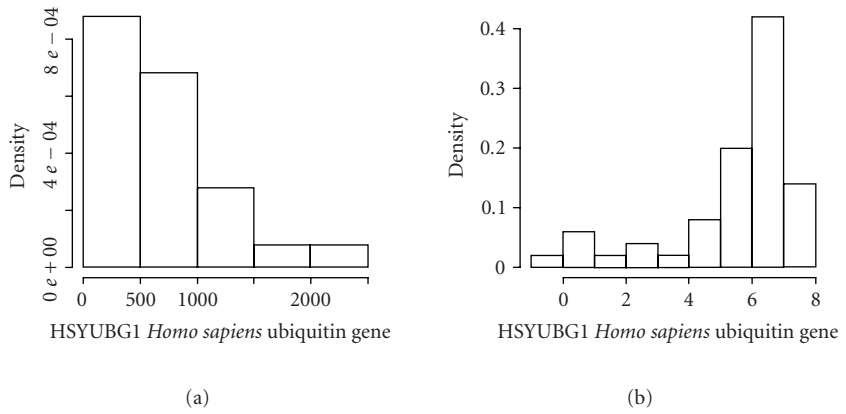


Figure 8.10. Distribution of expression data of the *HSYUBG1 Homo sapiens* ubiquitin gene in a data set of 50 prostatectomy samples measured with the U95Av2 Affymetrix microarray. (a) The histogram of the original expression data. (b) The histogram of the log-transformed gene expression data.

modelled by Gaussian distributions with linear dependencies. However, linearity of the parent-child dependencies and normality of the data are limitations. This section describes a new class of Bayesian networks that addresses these issues.

8.4.2.1. Learning and representation

A feature of gene expression data measured with microarray is the apparent lack of symmetry and there is evidence that they do not follow Gaussian distributions, even after a logarithmic transformation [85]. Figure 8.10 shows an example. The histogram in (a) shows the density of a sample of 50 expression levels of the *Homo sapiens* ubiquitin gene in the U95Av2 Affymetrix microarray. The distribution has an exponential decay, with a long right tail. The histogram in (b) displays the distribution of the log-transformed data and shows the phenomenon that log-transforming the original data removes the right tail but introduces a long left tail. This phenomenon is typically observed when log-transforming data that follow a Gamma distribution, with consequent bias induced to estimate the mean [86, Chapter 8]. We recently introduced a new class of Bayesian networks called generalized Gamma networks (GGN) able to describe possibly nonlinear dependencies between variables with nonnormal distributions [56]. Compared to other Bayesian network formalisms that have been proposed for representing gene-gene interactions [11], GGNs do not require to discretize gene expression data, or to enforce normality or log-normality assumptions.

In a GGN the conditional distribution of each variable Y_i given the parents $\mathrm{Pa}(y_i) = \{Y_{i1}, \ldots, Y_{ip(i)}\}$ follows a Gamma distribution $Y_i | \mathrm{pa}(y_i), \theta_i \sim \mathrm{Gamma}(\alpha_i, \mu_i(\mathrm{pa}(y_i), \beta_i))$, where $\mu_i(\mathrm{pa}(y_i), \beta_i)$ is the conditional mean of Y_i and $\mu_i(\mathrm{pa}(y_i), \beta_i)^2/\alpha_i$ is the conditional variance. We use the standard parameterization

Table 8.2. Link functions and parameterizations of the linear predictor.

Link	$g(\cdot)$	Linear predictor η
Identity	$\mu = \eta$	$\eta_i = \beta_{i0} + \sum_j \beta_{ij} y_{ij}$
Inverse	$\mu = \eta^{-1}$	$\eta_i = \beta_{i0} + \sum_j \beta_{ij} y_{ij}^{-1}$
Log	$\mu = e^{\eta}$	$\eta_i = \beta_{i0} + \sum_j \beta_{ij} \log(y_{ij})$

of generalized linear models [86], in which the mean $\mu_i(\text{pa}(y_i), \beta_i)$ is not restricted to be a linear function of the parameters β_{ij}, but the linearity in the parameters is enforced in the *linear predictor* η_i, which is itself related to the mean function by the *link function* $\mu_i = g(\eta_i)$. Therefore, we model the conditional density function as

$$p(y_i | \text{ pa }(y_i), \theta_i) = \frac{\alpha_i^{\alpha_i}}{\Gamma(\alpha_i)\mu_i^{\alpha_i}} y_i^{\alpha_i - 1} e^{-\alpha_i y_i / \mu_i}, \quad y_i \geq 0, \tag{8.38}$$

where $\mu_i = g(\eta_i)$ and the linear predictor η_i is parameterized as

$$\eta_i = \beta_{i0} + \sum_j \beta_{ij} f_j(\text{ pa}(y_i)) \tag{8.39}$$

and $f_j(\text{pa}(y_i))$ are possibly nonlinear functions. The linear predictor η_i is a function linear in the parameters β, but it is not restricted to be a linear function of the parent values, so that the generality of Gamma networks is in the ability to encode general nonlinear stochastic dependency between the node variables. Table 8.2 shows example of nonlinear mean functions. Figure 8.11 shows some examples of Gamma density functions, for different shape parameters $\alpha = 1, 1.5, 5$ and mean $\mu = 400$. Note that approximately symmetrical distributions are obtained for particular values of the shape parameter α.

Unfortunately, there is no closed form solution to learn the parameters of a GGN and we have therefore to resort to Markov chain Monte Carlo methods to compute stochastic estimates [20], or to maximum likelihood to compute numerical approximation of the posterior modes [87]. A well-known property of generalized linear models is that the parameters β_{ij} can be estimated independently of α_i, which is then estimated conditionally on β_{ij} [86].

To compute the maximum likelihood estimates of the parameters β_{ij} within each family $(Y_i, \text{Pa}(y_i))$, we need to solve the system of equations $\partial \log p(\mathcal{D}|\theta_i)/\partial\beta_{ij} = 0$. The Fisher scoring method is the most efficient algorithm to find the solution of the system of equations. This iterative procedure is a generalization of the Newton-Raphson procedure in which the Hessian matrix is replaced by its expected value. This modification speeds up the convergence rate of the iterative procedure that is known for being usually very efficient—it usually converges in 5 steps for appropriate initial values. Details can be found for example in [86].

Once the ML estimates of β_{ij} are known, say $\hat{\beta}_i$, we compute the fitted means $\hat{\mu}_{ik} = g(\hat{\beta}_{i0} + \sum_j \hat{\beta}_{ij} f_j(\text{pa}(y_i)))$ and use these quantities to estimate the shape parameter α_i. Estimation of the shape parameter in Gamma distributions is an

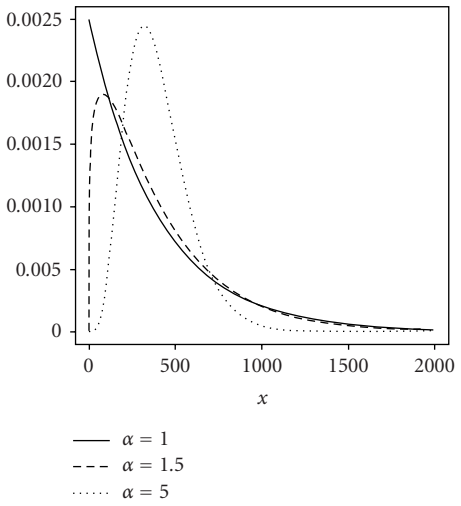


Figure 8.11. Example of Gamma density functions for shape parameters $\alpha = 1$ (continuous line), $\alpha = 1.5$ (dashed line), and $\alpha = 5$ (dotted line), and mean $\mu = 400$. For fixed mean, the parameter α determines the shape of the distribution that is skewed to the left for small α and approaches symmetry as α increases.

open issue, and authors have suggested several estimators (see, e.g., [86]). Popular choices are the deviance-based estimator that is defined as

$$\tilde{\alpha}_i = \frac{n-q}{\sum_k \left(y_{ik} - \hat{\mu}_{ik}\right)^2 / \hat{\mu}_{ik}^2}, \tag{8.40}$$

where q is the number of parameters β_{ij} that appear in the linear predictor. The maximum likelihood estimate $\hat{\alpha}_i$ of the shape parameter α_i would need the solution of the equation

$$n + n\log\left(\alpha_i\right) + n\frac{\Gamma\left(\alpha_i\right)'}{\Gamma\left(\alpha_i\right)} + -\sum_k \log\left(\hat{\mu}_{ik}\right) + \sum_k \log\left(y_{ik}\right) - \sum_i \frac{y_{ik}}{\hat{\mu}_{ik}} = 0 \tag{8.41}$$

with respect to α_i. We have an approximate closed form solution to this equation based on a Taylor expansion that is discussed in [68].

Also the model selection process requires the use of approximation methods. In this case, we use the Bayesian information criterion (BIC) [87] to approximate the marginal likelihood by $2\log p(\mathcal{D}|\hat{\theta}) - n_p \log(n)$, where $\hat{\theta}$ is the maximum likelihood estimate of θ and n_p is the overall number of parameters in the network. BIC is independent of the prior specification on the model space and trades off goodness of fit—measured by the term $2\log p(\mathcal{D}|\hat{\theta})$—and model complexity—measured by the term $n_p \log(n)$. We note that BIC factorizes into a product of terms for each variable Y_i and makes it possible to conduct local structural learning.

Table 8.3. The nine genes used in the GGN and their known functions.

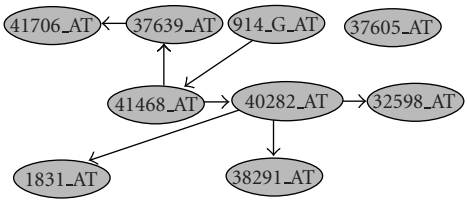
Affy-entry	Gene name	Gene function
41706_At	*alpha-methylacyl-CoA racemase*	Cellular component
37639_At	*Hepsin*	Cell growth
37605_At	*COL2A1*	*Collagen*
41468_At	TCRγ	Cellular defense
914_G_At	*ERG*	Transcription regulation
40282_S_At	*Adipsin*	Role in immune system biology
1831_At	*TGFβ*	Transforming grown factor
38291_At	Human *enkephalin* gene	Signal transduction
32598_A	*Nel-like 2*	Cell growth regulation and differentiation

While the general type of dependencies in Gamma networks makes it possible to model a variety of dependencies within the variables, exact probabilistic reasoning with the network becomes impossible and we need to resort to Gibbs sampling (see Section 8.2). Our simulation approach uses the adaptative rejection metropolis sampling (ARMS) of [88] when the conditional density $p(y_i|\mathcal{Y}\backslash y_i, \hat{\theta})$ is log-concave, and adaptive rejection with metropolis sampling in the other cases. See [56] for more details.

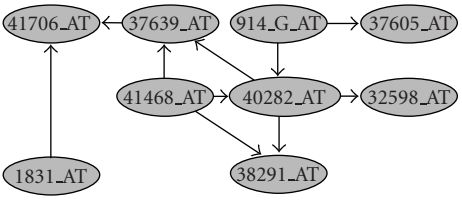
8.4.2.2. An example

We use a subset of nine of the gene expression data measured from the 102 prostatectomy specimens to show the representation advantages of GGNs. The nine genes are listed in Table 8.3. We modeled the dependency structure among the nine genes in the normal and tumor specimens, with an initial order that was chosen by using information about their roles in pathways, when known, and by ranking the remaining genes on the basis of the evidence for differential expression. For example, the gene 914_G_AT (ERG) has a transcription regulation function that has been observed in several tumors, so we left this gene high in the order and tested it as a parent of all the other nodes. Figure 8.12 depicts the dependency structures in the two groups. In both cases, we limited the search to dependency models in which the link function was either the identity $\mu = \eta$ or the inverse link $\mu = 1/\eta$. The two network structures were validated by examining the blanket residuals to assess the goodness of fit for each local dependency structure. In both networks we tested whether the standardized blanket residuals had means significantly different from 0 using standard t-tests, and we checked for departures from normality. These tests confirmed the validity of the network structures induced from data, and the correctness of the distributional assumptions.

Evidence of the biological meaning of the dependency structures encoded by the two GGNs gives further support that this technology can help to model complex gene-gene interactions in biological systems. For example, in the network learned from the normal specimens, the gene COL2A1 (37605_AT: a collagene) is independent of all other genes, whereas in the network learned from the tumor specimens, this gene is a child of ERG (914_AT: an oncogene with transcription



(a)



(b)

Figure 8.12. Gamma networks induced from (a) the 50 normal specimens and (b) the 52 tumor specimens (right).

regulation functions). Independent studies have associated changes of expression in TGFβ (1831_AT: a gene with role in signalling pathways), with changes of expression in COL2A1, and our models suggest a possible mechanism in which this occurs. In the network induced from tumor specimens, TGFβ is directly influencing AMACR (41706_AT: a gene known as a marker of tumor differentiation). In both networks, the dependency structure of Adipsin (40282_S_AT: a gene supposed to have a role in immune system biology) is essentially the same, besides the fact that Epsin (37639_AT: a gene with putative function in cell growth) is independent of Adipsin given TCRγ (41468_AT: a gene with role in cell defense) in the network learned from normal specimens. However, even for those genes with the same dependency structure, the probability distributions that quantify these dependencies suggest different gene-gene interactions. Figure 8.13 shows the smooth, nonlinear dependency between Adipsin and Nel-like 2 (32598_A) in the two GGNs induced from (a) the 50 normal specimens and (b) the 52 tumor specimens. The two nonlinear dependencies show that changes of expression of Adipsin in the network learned from tumor specimens have a much reduced effect on changes of expression of Nel-like 2. As mentioned earlier, one theory is that cancer arises from the accumulation of genetic changes that induce unlimited, self-sufficient growth and resistance to normal regulatory mechanisms. Our different dependency structures suggest that, in the cancer specimens, the gene Adipsin has a weaker control on the gene Nel-like 2 that regulates cell growth and differentiation. The reasonable biological explanation also points out

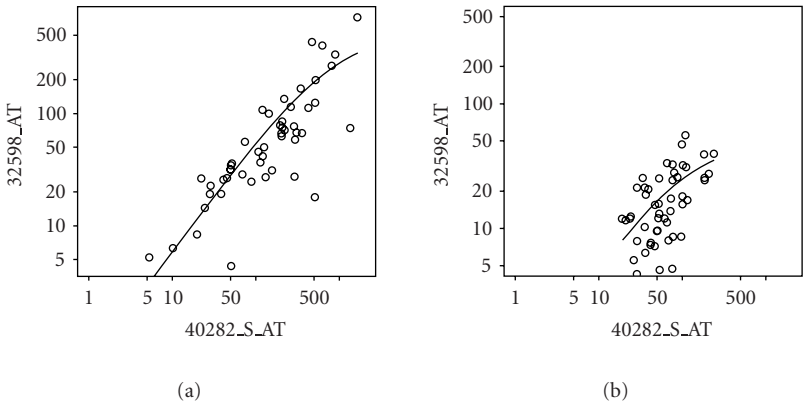


Figure 8.13. Scatter plot of the dependency between 40282_S_AT and 32598_A in the two GGNs induced from (a) the 50 normal specimens and (b) the 52 tumor specimens. The lines are the dependency fitted by the GGNs. Both plots are in log-scales.

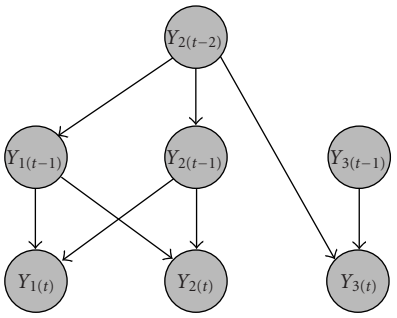
an important feature of GGNs: by modeling gene-gene interaction via nonlinear dependency, GGNs can easily describe the biological effect of gene expression saturation, in which gene expression control changes according to changes of expression levels.

Alternative dependency structures can suggest new hypothetical pathways as well as experiments to test putative functions of genes. For example, the propagation of particular expression levels for some genes can identify their impact on the expression level of other genes and provide a platform for *in silico* experiments based on the learned network.

8.4.3. Bayesian networks and temporal dependency

One of the limitations of Bayesian networks is the inability to represent forward loops: by definition, the directed graph that encodes the marginal and conditional independencies between the network variables cannot have cycles. This limitation makes traditional Bayesian networks unsuitable for the representation of many biological systems in which feedback controls are a critical aspect of gene regulation. Dynamic Bayesian networks provide a general framework to integrate multivariate time series of gene products and to represent feed-forward loops and feedback mechanisms [11] that are alternative to other network models of gene regulation [89].

A dynamic Bayesian network is defined by a directed acyclic graph in which nodes continue to represent stochastic variables and arrows represent temporal dependencies that are quantified by probability distributions. The crucial assumption is that the probability distributions of the temporal dependencies are time invariant, so that the directed acyclic graph of a dynamic Bayesian network represents only the necessary and sufficient time transitions to reconstruct the overall



		$Y_{3(t)}$	
$Y_{2(t-2)}$	$Y_{3(t-1)}$	−	+
−	−	0.9	0.1
−	+	0.7	0.3
+	−	0.6	0.4
+	+	0.1	0.9

Figure 8.14. A directed acyclic graph that represents the temporal dependency of three categorical variables describing up (+) and down (−) regulations of three genes.

temporal process. Figure 8.14 shows the directed acyclic graph of a dynamic Bayesian network with three variables. The subscript of each node denotes the time lag, so that the arrows from the nodes $Y_{2(t-1)}$ and $Y_{1(t-1)}$ to the node $Y_{1(t)}$ describe the dependency of the probability distribution of the variable Y_1 at time t on the value of Y_1 and Y_2 at time $t-1$. Similarly, the directed acyclic graph shows that the probability distribution of the variable Y_2 at time t is a function of the value of Y_1 and Y_2 at time $t-1$. This symmetrical dependency allows us to represent feedback loops and we used it to describe the regulatory control of glucose in diabetic patients [90]. A dynamic Bayesian network is not restricted to represent temporal dependency of order 1. For example the probability distribution of the variable Y_3 at time t depends on the value of the variable at time $t-1$ as well as the value of the variable Y_2 at time $t-2$. The conditional probability table in Figure 8.14 shows an example when the variables Y_2, Y_3 are categorical.

By using the local Markov property, the joint probability distribution of the three variables at time t, given the past history $y_{1(t-1)},\ldots,y_{1(t-l)}, y_{2(t-1)},\ldots,y_{2(t-l)}, y_{3(t-1)},\ldots,y_{3(t-l)}$ is given by the product of the three factors:

$$\begin{aligned}
&p\left(y_{1(t)}|y_{1(t-1)},\ldots,y_{1(t-l)},y_{2(t-1)},\ldots,y_{2(t-l)},y_{3(t-1)},\ldots,y_{3(t-l)}\right)\\
&\qquad = p\left(y_{1(t)}|y_{1(t-1)},y_{2(t-1)}\right),\\
&p\left(y_{2(t)}|y_{1(t-1)},\ldots,y_{1(t-l)},y_{2(t-1)},\ldots,y_{2(t-l)},y_{3(t-1)},\ldots,y_{3(t-l)}\right)\\
&\qquad = p\left(y_{2(t)}|y_{1(t-1)},y_{2(t-1)}\right),\\
&p\left(y_{3(t)}|y_{1(t-1)},\ldots,y_{1(t-l)},y_{2(t-1)},\ldots,y_{2(t-l)},y_{3(t-1)},\ldots,y_{3(t-l)}\right)\\
&\qquad = p\left(y_{3(t)}|y_{3(t-1)},y_{2(t-2)}\right)
\end{aligned} \tag{8.42}$$

that represents the probability of transition over time. By assuming that these probability distributions are time invariant, they are sufficient to compute the probability that a process that starts from known values $y_{1(1)}$, $y_{2(1)}$, $y_{3(0)}$, $y_{3(1)}$ evolves into $y_{1(T)}$, $y_{2(T)}$, $y_{3(T)}$ by using one of the algorithms for probabilistic reasoning described in Section 8.2. The same algorithms can be used to compute the

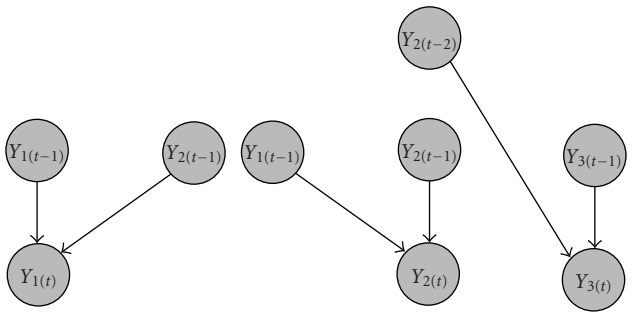


Figure 8.15. Modular learning of the dynamic Bayesian network in Figure 8.14. First a regressive model is learned for each of the three variables at time t, and then the three models are joined by their common ancestors $Y_{1(t-1)}$ and $Y_{2(t-2)}$ to produce the directed acyclic graph in Figure 8.14.

probability that a process with values $y_{1(T)}$, $y_{2(T)}$, $y_{3(T)}$ at time T started from the initial states $y_{1(1)}$, $y_{2(1)}$, $y_{3(0)}$, $y_{3(1)}$.

Learning dynamic Bayesian networks when all the variables are observable is a straightforward parallel application of the structural learning described in Section 8.2.2. To build the network, we proceed by selecting the set of parents for each variable Y_i at time t, and then the models are joined by the common ancestors. An example is in Figure 8.15. The search of each local dependency structure is simplified by the natural ordering imposed on the variables by the temporal frame [91] that constrains the model space of each variable Y_i at time t: the set of candidate parents consists of the variables $Y_{i(t-1)}, \ldots, Y_{i(t-p)}$ as well as the variables $Y_{h(t-j)}$ for all $h \neq i$ and $j = 1, \ldots, p$. The K2 algorithm [43] discussed in Section 8.2.2 appears to be particularly suitable for exploring the space of dependency for each variable $Y_{i(t)}$. The only critical issue is that the selection of the largest temporal order to explore depends on the sample size, because each temporal lag of order p leads to the loss of the first p temporal observations in the data set [55].

Dynamic Bayesian networks are an alternative approach to represent gene regulatory mechanisms by approximating rates of change described by a system of differential equations with autoregressive models. When the gene products are measured at regularly spaced time points, there is a simple way to approximate the rate of change $dy_{i(t)}/dt = f(y_{gt})$ by a first order linear approximation. This approach has been used to model the rate of change by linear Gaussian networks [92]. However, the development of similar approximations for nonregularly spaced time points and for general, nonlinear, kinetic equations with feedback loops [93] is an open issue. The further advantage of dynamic Bayesian network is to offer an environment for causal inference with well-designed temporal experiments.

8.5. Research directions

This chapter has discussed the potential usefulness of Bayesian networks to analyze genomic data. However, there are some limitations of the current representation

and learning approaches that need further investigation. A main assumption underlying all learning algorithms is that the data are complete, so there are no missing entries and both gene expression data measured with cDNA microarrays and genotype data have missing values. Furthermore, often some of the variables in the data set are continuous and some are discrete and to use standard algorithms for learning a Bayesian network from data, the continuous variables are usually discretized with potential loss of information.

Mixed variable networks. The process of learning Bayesian networks requires two components: a search procedure to scan through a set of possible models and a scoring metric, such as BIC or the marginal likelihood, to select one of the models. We have shown in Section 8.2.2 that when the variables in the network are all continuous, a closed-form solution for the calculation of the marginal likelihood exists under the assumption that each variable is normally distributed around a mean, which *linearly* depends on its parent variables [36, 94]. The drawback is that they are heavily limited in their representation power, as they can only capture linear dependencies among continuous variables. To increase their scope, Gaussian linear networks have been extended into a mixture of Gaussian networks, which model a conditional distribution as a weighted mixture of linear Gaussian distributions and can, in principle, represent a wider variety of interactions. Unfortunately, no closed-form solution exists to compute the marginal likelihood of these distributions, and we have to resort to computationally demanding approximation methods [95]. The normality assumption on the variables can be relaxed to the more general case that the variables have distributions in the exponential family, and we have introduced the family of GGNs to describe dependency structures of nonnormal variables with possibly nonlinear dependencies. The crucial assumption in GGNs is that all variables in the network have probability distributions in the same family. An important and yet unsolved issue is the learning of mixed networks, in which some variables are continuous and some are discrete. Imposing the assumption that discrete variables can only be parent nodes in the network but cannot be children of any continuous Gaussian node leads to a closed form solution for the computation of the marginal likelihood [96]. This property has been applied, for example, to model-based clustering in [97], and it is commonly used in classification problems [98]. However, this restriction can quickly become unrealistic and greatly limits the set of models to explore. As a consequence, common practice is still to discretize continuous variables with possible loss of information, particularly when the continuous variables are highly skewed.

Missing data. The received view of the effect of missing data on statistical inference is based on the approach described by Rubin in [99]. This approach classifies the missing data mechanism as ignorable or not, according to whether the data are missing completely at random (MCAR), missing at random (MAR), or informatively missing (IM). According to this approach, data are MCAR if the probability that an entry is missing is independent of both observed and unobserved values. They are MAR if this probability is at most a function of the observed values in the database and, in all other cases, data are IM. The received view is that, when data

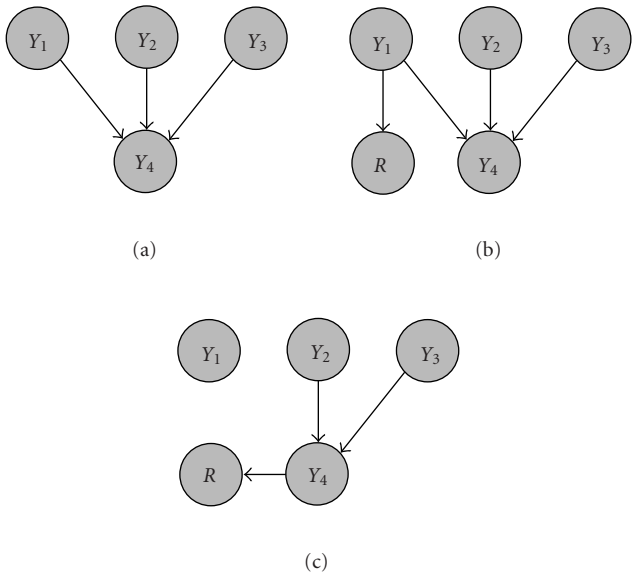


Figure 8.16. An example of partially ignorable missing data mechanism. (a) The variable Y_4 in the Bayesian network is only partially observed, while the parents Y_1, Y_2, Y_3 are fully observed. (b) The variable R encodes whether Y is observed ($R = 1$) or not ($R = 0$). Because the variable R is a child of Y_1, which is fully observed, data are MAR. (c) Removing the variable Y_1 from the dependency model for Y_4 induces a link between Y and R so that the missing data mechanism becomes informative.

are either MCAR or MAR, the missing data mechanism is ignorable for parameter estimation, but it is not when data are IM.

An important but overlooked issue is whether the missing data mechanism generating data that are MAR is ignorable for model selection [100, 101]. We have shown that this is not the case for the class of graphical models exemplified in Figure 8.16 [101]. We assume that there is only one variable with missing data (the variable Y_4 in the DAG) and that its possible parents are all fully observed. To model the missing data mechanism, we introduce the dummy variable R that takes on one of the two values: $R = 1$ when Y_4 is observed, and $R = 0$ when Y_4 is missing. The missing data mechanism can be described by the graphical structure relating R, Y_4 and Y_1, Y_2, Y_3: when R is not linked to any of the variables, data are MCAR; when R is linked to any subset of Y_1, Y_2, Y_3 but not Y_4, data are MAR; when R is linked to Y_4, data are IM. If the graphical structure is known, the missing data mechanism is ignorable for parameter estimation in the first two cases. However, when the task is to learn the graphical structure from data, only a mechanism generating data that are MCAR is ignorable. This fact is shown in Figure 8.16c: when we assess the dependency of Y_4 on Y_2, Y_3 but not Y_1, the variable R is linked to Y_4 so that the missing data mechanism is informative for this model structure.

We defined this mechanism only partially ignorable for model selection and we showed how to discriminate between ignorable and partially ignorable missing data mechanisms [101]. We also introduced two approaches to model selection with partially ignorable missing data mechanisms: *ignorable imputation* and *model folding*. Contrary to standard imputation schemes [102, 103, 104, 105], ignorable imputation accounts for the missing-data mechanism and produces, asymptotically, a proper imputation model as defined by Rubin [99, 106]. However, the computation effort can be very demanding and model folding is a deterministic method to approximate the exact marginal likelihood that reaches high accuracy at a low computational cost, because the complexity of the model search is not affected by the presence of incomplete cases. Both ignorable imputation and model folding reconstruct a completion of the incomplete data by taking into account the variables responsible for the missing data. This property is in agreement with the suggestion put forward in [103, 107, 108] that the variables responsible for the missing data should be kept in the model. However, our approach allows us to also evaluate the likelihoods of models that do not depend explicitly on these variables.

Although this work provides the analytical foundations for a proper treatment of missing data when the inference task is model selection, it is limited to the very special situation in which only one variable is partially observed, data are supposed to be only MCAR or MAR, and the set of Bayesian networks is limited to those in which the partially observed variable is a child of the other variables. Research is needed to extend these results to the more general graphical structures, in which several variables can be partially observed and data can be MCAR, MAR, or IM.

Acknowledgments

This research was supported by the National Science Foundation (0120309), the Spanish State Office of Education and Universities, the European Social Fund, and the Fulbright Program. We thank the reviewers and editors for their invaluable comments that helped improve the original version of this chapter.

Bibliography

[1] F. S. Collins, M. S. Guyer, and A. Charkravarti, "Variations on a theme: cataloging human DNA sequence variation," *Science*, vol. 278, no. 5343, pp. 1580–1581, 1997.

[2] E. S. Lander, "The new genomics: global views of biology," *Science*, vol. 274, no. 5287, pp. 536–539, 1996.

[3] W. W. Cai, J. H. Mao, C. W. Chow, S. Damani, A. Balmain, and A. Bradley, "Genome-wide detection of chromosomal imbalances in tumors using BAC microarrays," *Nat. Biotechnol.*, vol. 20, no. 4, pp. 393–396, 2002.

[4] D. J. Lockhart, H. Dong, M. C. Byrne, et al., "Expression monitoring by hybridization to high-density oligonucleotide arrays," *Nat. Biotechnol.*, vol. 14, no. 13, pp. 1675–1680, 1996.

[5] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[6] P. Sebastiani, E. Gussoni, I. S. Kohane, and M. F. Ramoni, "Statistical challenges in functional genomics," *Statist. Sci.*, vol. 18, no. 1, pp. 33–70, 2003, with comments by H. V. Baker and G. A. Churchill, and a rejoinder by the authors.

[7] E. S. Lander, "Array of hope," *Nat. Genet.*, vol. 21, no. Suppl 1, pp. 3–4, 1999.
[8] E. Phizicky, P. I. H. Bastiaens, H. Zhu, M. Snyder, and S. Fields, "Protein analysis on a proteomic scale," *Nature*, vol. 422, no. 6928, pp. 208–215, 2003.
[9] R. Aebersold and M. Mann, "Mass spectrometry-based proteomics," *Nature*, vol. 422, no. 6928, pp. 198–207, 2003.
[10] N. Freimer and C. Sabatti, "The human phenome project," *Nat. Genet.*, vol. 34, no. 1, pp. 15–21, 2003.
[11] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Science*, vol. 303, no. 5659, pp. 799–805, 2004.
[12] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian network to analyze expression data," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 601–620, 2000.
[13] D. Pe'er, A. Regev, G. Elidan, and N. Friedman, "Inferring subnetworks from perturbed expression profiles," *Bioinformatics*, vol. 17, no. Suppl 1, pp. S215–S224, 2001.
[14] E. Segal, M. Shapira, A. Regev, et al., "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," *Nat. Genet.*, vol. 34, no. 2, pp. 166–176, 2003.
[15] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller, "Rich probabilistic models for gene expression," *Bioinformatics*, vol. 17, no. Suppl 1, pp. S243–S252, 2001.
[16] R. Jansen, H. Yu, D. Greenbaum, et al., "A Bayesian networks approach for predicting protein-protein interactions from genomic data," *Science*, vol. 302, no. 5644, pp. 449–453, 2003.
[17] Z. Cai, E. F. Tsung, V. D. Marinescu, M. F. Ramoni, A. Riva, and I. S. Kohane, "Bayesian approach to discovering pathogenic SNPs in conserved protein domains," *Hum. Mutat.*, vol. 24, no. 2, pp. 178–184, 2004.
[18] P. Sebastiani, M. F. Ramoni, V. Nolan, C. T. Baldwin, and M. H. Steinberg, "Multigenic dissection and prognostic modeling of overt stroke in sickle cell anemia," *Blood*, vol.104, no. 11, pp. 460a, 2004.
[19] D. Heckerman, "Bayesian networks for data mining," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 79–119, 1997.
[20] D. Madigan and G. Ridgeway, "Bayesian data analysis for data mining," in *Handbook of Data Mining*, pp. 103–132, MIT Press, Cambridge, Mass, USA, 2003.
[21] D. Madigan and J. York, "Bayesian graphical models for discrete data," *Int. Statist. Rev.*, vol. 63, pp. 215–232, 1995.
[22] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*, John Wiley & Sons, Chichester, UK, 1990.
[23] A. P. Dawid, "Conditional independence in statistical theory," *J. Roy. Statist. Soc. Ser. B*, vol. 41, no. 1, pp. 1–31, 1979.
[24] A. P. Dawid, "Conditional independence for statistical operations," *Ann. Statist.*, vol. 8, no. 3, pp. 598–617, 1980.
[25] S. L. Lauritzen, *Graphical Models*, vol. 17 of *Oxford Statistical Science Series*, The Clarendon Press, Oxford University Press, New York, NY, USA, 1996.
[26] E. Castillo, J. M. Gutierrez, and A. S. Hadi, *Expert Systems and Probabilistic Network Models*, Springer, New York, NY, USA, 1997.
[27] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *J. Roy. Statist. Soc. Ser. B*, vol. 50, no. 2, pp. 157–224, 1988.
[28] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, Calif, USA, 1988.
[29] J. Cheng and M. J. Druzdzel, "AIS-BN: an adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks," *J. Artificial Intelligence Res.*, vol. 13, pp. 155–188, 2000.
[30] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, 1984.

[31] A. Thomas, D. J. Spiegelhalter, and W. R. Gilks, "Bugs: a program to perform Bayesian inference using Gibbs sampling," in *Bayesian Statistics 4*, J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, Eds., pp. 837–842, The Clarendon Press, Oxford University Press, New York, NY, USA, 1992.

[32] M. F. Ramoni and P. Sebastiani, "Bayesian methods for intelligent data analysis," in *Intelligent Data Analysis: An Introduction*, pp. 131–168, Springer, New York, NY, USA, 2nd edition, 2003.

[33] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, vol. 81 of *Lecture Notes in Statistics*, Springer, New York, NY, USA, 1993.

[34] A. P. Dawid and S. L. Lauritzen, "Hyper-Markov laws in the statistical analysis of decomposable graphical models," *Ann. Statist.*, vol. 21, no. 3, pp. 1272–1317, 1993.

[35] A. P. Dawid and S. L. Lauritzen, "Correction: "Hyper-Markov laws in the statistical analysis of decomposable graphical models" [Ann. Statist. **21** (1993), no. 3, 1272–1317]," *Ann. Statist.*, vol. 23, no. 5, p. 1864, 1995.

[36] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: the combinations of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995.

[37] D. J. Spiegelhalter and S. L. Lauritzen, "Sequential updating of conditional probabilities on directed graphical structures," *Networks*, vol. 20, no. 5, pp. 579–605, 1990.

[38] I. J. Good, *The Estimation of Probabilities. An Essay on Modern Bayesian Methods*, MIT Press, Cambridge, Mass, USA, 1968.

[39] D. Geiger and D. Heckerman, "A characterization of Dirichlet distributions through local and global independence," *Ann. Statist.*, vol. 25, pp. 1344–1368, 1997.

[40] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer, New York, NY, USA, 1999.

[41] D. Geiger and D. Heckerman, "Learning Gaussian networks," in *Proc. 10th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 235–243, Morgan Kaufmann, San Mateo, Calif, USA, 1994.

[42] A. O'Hagan, *Kendall's Advanced Theory of Statistics, Bayesian Inference*, Edward Arnold, London, UK, 1994.

[43] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.

[44] P. Larranaga, C. Kuijpers, R. Murga, and Y. Yurramendi, "Learning Bayesian network structures by searching for the best ordering with genetic algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 4, pp. 487–493, 1996.

[45] M. Singh and M. Valtorta, "Construction of Bayesian network structures from data: a brief survey and an efficient algorithm," *Int. J. Approx. Reasoning*, vol. 12, no. 2, pp. 111–131, 1995.

[46] H. Zhou and S. Sakane, "Sensor planning for mobile robot localization using Bayesian network inference," *Advanced Robotics*, vol. 16, no. 8, pp. 751–771, 2002.

[47] P. Sebastiani, M. F. Ramoni, and A. Crea, "Profiling your customers using Bayesian networks," *ACM SIGKDD Explorations*, vol. 1, no. 2, pp. 91–96, 2000.

[48] P. Sebastiani and M. F. Ramoni, "On the use of Bayesian networks to analyze survey data," *Research in Official Statistics*, vol. 4, no. 1, pp. 53–64, 2001.

[49] N. Friedman and D. Koller, "Being Bayesian about network structure: a Bayesian approach to structure discovery in Bayesian networks," *Machine Learning*, vol. 50, no. 1-2, pp. 95–125, 2003.

[50] S. W. Doniger, N. Salomonis, K. D. Dahlquist, K. Vranizan, S. C. Lawlor, and B. R. Conklin, "MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data," *Genome Biol.*, vol. 4, no. 1, pp. R7, 2003.

[51] K. D. Dahlquist, N. Salomonis, K. Vranizan, S. C. Lawlor, and B. R. Conklin, "GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways," *Nat. Genet.*, vol. 31, no. 1, pp. 19–20, 2002.

[52] D. M. Chickering, "Learning equivalence classes of Bayesian-network structures," *J. Mach. Learn. Res.*, vol. 2, no. 3, pp. 445–498, 2002.

[53] S. G. Bøttcher and C. Dethlefsen, "Deal: a package for learning Bayesian networks," http://www.jstatsoft.org/v08/i20/.

[54] D. Madigan and A. E. Raftery, "Model selection and accounting for model uncertainty in graphical models using Occam's window," *J. Amer. Statist. Assoc.*, vol. 89, no. 428, pp. 1535–1546, 1994.

[55] J. Yu, V. Smith, P. Wang, A. Hartemink, and E. Jarvis, "Using Bayesian network inference algorithms to recover molecular genetic regulatory networks," in *Proc. 3rd International Conference on Systems Biology*, Stockholm, Sweden, December 2002.

[56] P. Sebastiani and M. F. Ramoni, "Generalized gamma networks," Tech. Rep., Department of Mathematics and Statistics, University of Massachusetts, Amhest, Mass, USA, 2003.

[57] I. J. Good, "Rational decisions," *J. Roy. Statist. Soc. Ser. B.*, vol. 14, pp. 107–114, 1952.

[58] D. J. Hand, *Construction and Assessment of Classification Rules*, John Wiley & Sons, New York, NY, USA, 1997.

[59] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, USA, 2001.

[60] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification," *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[61] C. Yoo, V. Thorsson, and G. F. Cooper, "Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data," *Pac. Symp. Biocomput.*, vol. 7, pp. 498–509, 2002.

[62] J. M. Schildkraut, "Examining complex genetic interactions," in *Approaches to Gene Mapping in Complex Human Diseases*, pp. 379–410, John Wiley & Sons, New York, NY, USA, 1998.

[63] D. Altshuler, J. N. Hirschhorn, M. Klannemark, et al., "The common PPARγ Pro12Ala polymorphism is associated with decreased risk of type 2 diabetes," *Nat. Genet.*, vol. 26, no. 1, pp. 76–80, 2000.

[64] J. Ott, *Analysis of Human Genetic Linkage*, Johns Hopkins University Press, Baltimore, Md, USA, 1999.

[65] N. P. Jewell, *Statistics for Epidemiology*, CRC/Chapman and Hall, Boca Raton, Fla, USA, 2003.

[66] P. Sebastiani, M. F. Ramoni, V. Nolan, C. T. Baldwin, and M. H. Steinberg, "Genetic dissection and prognostic modeling of a complex trait: overt stroke in sickle cell anemia," to appear in Natural Genetics.

[67] D. Singh, P. G. Febbo, K. Ross, et al., "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

[68] P. Sebastiani, M. F. Ramoni, and I. Kohane, "BADGE: technical notes," Tech. Rep., Department of Mathematics and Statistics, University of Massachusetts, Amhest, Mass, USA, 2003.

[69] S. M. Dhanasekaran, T. R. Barrette, D. Ghosh, et al., "Delineation of prognostic biomarkers in prostate cancer," *Nature*, vol. 412, no. 6849, pp. 822–826, 2001.

[70] M. Essand, G. Vasmatzis, U. Brinkmann, P. Duray, B. Lee, and I. Pastan, "High expression of a specific T-cell receptor γ transcript in epithelial cells of the prostate," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 16, pp. 9287–9292, 1999.

[71] P. Spirtes, C. Glymour, and R. Scheines, "Constructing Bayesian network models of gene expression networks from microarray data," in *Proc. Atlantic Symposium on Computational Biology, Genome Information Systems & Technology*, Durham, NC, USA, March 2001.

[72] R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[73] G. Parmigiani, E. S. Garrett, R. Anbazhagan, and E. Gabrielson, "A statistical framework for expression-based molecular classification in cancer," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 64, no. 4, pp. 717–736, 2002.

[74] M. West, C. Blanchette, H. Dressman, et al., "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 20, pp. 11462–11467, 2001.

[75] A. D. Keller, M. Schummer, L. Hood, and W. L. Ruzzo, "Bayesian classification of DNA array expression data," Tech. Rep. UW-CSE-2000-08-01, Department of Computer Science and Engineering, University of Washington, Seattle, Wash, USA, 2000.

[76] Y. Zhu, J. Hollmen, R. Raty, et al., "Investigatory and analytical approaches to differential gene expression profiling in mantle cell lymphoma," *Br. J. Haematol.*, vol. 119, no. 4, pp. 905–915, 2002.

[77] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, NY, USA, 1973.

[78] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proc. 10th National Conference on Artificial Intelligence*, pp. 223–228, AAAI Press, Menlo Park, Calif, USA, 1992.

[79] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2, pp. 131–163, 1997.

[80] M. Sahami, "Learning limited dependence Bayesian classifiers," in *Proc. 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 334–338, AAAI Press, Menlo Park, Calif, USA, 1996.

[81] A. Ben Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," in *Proc. 4th Annual International Conference on Computational Molecular Biology*, pp. 54–64, Tokyo, Japan, 2000.

[82] Y. Li, C. Campbell, and M. Tipping, "Bayesian automatic relevance determination algorithms for classifying gene expression data," *Bioinformatics*, vol. 18, no. 10, pp. 1332–1339, 2002.

[83] B. Krishnapuram, L. Carin, and A. J. Hartemink, "Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data," *J. Comput. Biol.*, vol. 11, no. 2-3, pp. 227–242, 2004.

[84] P. Langley and S. Sage, "Induction of selective Bayesian classifiers," in *Proc. 10th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 399–406, Morgan Kaufmann, San Mateo, Calif, USA, 1994.

[85] P. Sebastiani, J. Jeneralczuk, and M. F. Ramoni, "Design and analysis of screening experiments with microarrays," in *Screening*, A. Dean and S. Lewis, Eds., Springer, New York, NY, USA, in press.

[86] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, Chapman and Hall, London, UK, 2nd edition, 1989.

[87] R. E. Kass and A. Raftery, "Bayes factors," *J. Amer. Statist. Assoc.*, vol. 90, no. 430, pp. 773–795, 1995.

[88] W. R. Gilks and G. O. Roberts, "Strategies for improving MCMC," in *Markov Chain Monte Carlo in Practice*, W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds., pp. 89–114, Chapman and Hall, London, UK, 1996.

[89] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[90] M. F. Ramoni, A. Riva, M. Stefanelli, and V. Patel, "An ignorant belief network to forecast glucose concentration from clinical databases," *Artif. Intell. Med.*, vol. 7, no. 6, pp. 541–559, 1995.

[91] N. Friedman, K. Murphy, and S. Russell, "Learning the structure of dynamic probabilistic networks," in *Proc. 14th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 139–147, Morgan Kaufmann, San Mateo, Calif, USA, 1998.

[92] M. J. de Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano, "Inferring gene regulatory networks from time-ordered gene expression data of Bacillus subtilis using differential equations," *Pac. Symp. Biocomput.*, vol. 8, pp. 17–28, 2003.

[93] T. Chen, H. L. He, and G. M. Church, "Modeling gene expression with differential equations," *Pac. Symp. Biocomput.*, vol. 4, pp. 29–40, 1999.

[94] B. Thiesson, "Accelerated quantification of Bayesian networks with incomplete data," in *Proc. 1st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 306–311, ACM Press, New York, NY, USA, 1995.

[95] D. M. Chickering and D. Heckerman, "Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables," *Machine Learning*, vol. 29, no. 2-3, pp. 181–212, 1997.

[96] S. L. Lauritzen, "Propagation of probabilities, means, and variances in mixed graphical association models," *J. Amer. Statist. Assoc.*, vol. 87, no. 420, pp. 1098–1108, 1992.

[97] M. F. Ramoni, P. Sebastiani, and I. S. Kohane, "Cluster analysis of gene expression dynamics," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 14, pp. 9121–9126, 2002.
[98] P. Cheeseman and J. Stutz, "Bayesian classification (AutoClass): Theory and results," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds., pp. 153–180, MIT Press, Cambridge, Mass, USA, 1996.
[99] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*, John Wiley & Sons, New York, NY, USA, 1987.
[100] D. B. Rubin, "Multiple imputation after 18 years," *J. Amer. Statist. Assoc.*, vol. 91, no. 434, pp. 473–489, 1996.
[101] P. Sebastiani and M. F. Ramoni, "Bayesian selection of decomposable models with incomplete data," *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1375–1386, 2001.
[102] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman and Hall, London, UK, 1995.
[103] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, John Wiley & Sons, New York, NY, USA, 1987.
[104] J. L. Schafer, *Analysis of Incomplete Multivariate Data*, vol. 72 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London, UK, 1997.
[105] Y. Thibaudeau and W. E. Winler, "Bayesian networks representations, generalized imputation, and synthetic microdata satisfying analytic constraints," Tech. Rep. RRS2002/09, Statistical Research Division, U.S. Census Bureau, Washington, DC, USA, 2002, http://www.census.gov/srd/www/byyear.html.
[106] D. B. Rubin, H. S. Stern, and V. Vehovar, "Handling "don't know" survey responses: the case of the Slovenian plebiscite," *J. Amer. Statist. Assoc.*, vol. 90, no. 431, pp. 822–828, 1995.
[107] D. F. Heitjan and D. B. Rubin, "Ignorability and coarse data," *Ann. Statist.*, vol. 19, no. 4, pp. 2244–2253, 1991.
[108] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.
[109] Z. Cai, E. F. Tsung, V. D. Marinescu, M. F. Ramoni, A. Riva, and I. S. Kohane, "Bayesian approach to discovering pathogenic SNPs in conserved protein domains," *Hum. Mutat.*, vol. 24, no. 2, pp. 178–184, 2004.
[110] M. A. Tanner, *Tools for Statistical Inference*, Springer, New York, NY, USA, 3rd edition, 1996.

Paola Sebastiani: Department of Biostatistics, School of Public Health, Boston University, 715 Albany Street, Boston, MA 02118, USA

Email: sebas@bu.edu

Maria M. Abad: Software Engineering Department, University of Granada, Daniel Saucedo Aranda, Granada 18071, Spain

Email: mabad@ugr.es

Marco F. Ramoni: Pediatrics and Medicine, Harvard Medical School, Bioinformatics, Harvard Partners Genome Center, Children's Hospital Informatics Program, 300 Longwood Avenue, Boston, MA 02115, USA

Email: marco_ramoni@harvard.edu

9 Statistical inference of transcriptional regulatory networks

Xiaodong Wang, Dimitris Anastassiou, and Dong Guo

We give a general overview of modeling of gene regulatory networks and discuss various statistical inference problems related to these models. First various gene function modeling techniques are described, including qualitative models such as directed and undirected graphs, Boolean networks, and logic networks, and quantitative models, including differential equations, linear and nonlinear function models, and radial basis functions. Then parameter estimation methods are discussed for known network structures, including equation-based methods and Bayesian methods. Finally, Bayesian techniques for inferring network structures are discussed.

9.1. Introduction

A central theme of molecular biology is to understand the regulatory mechanism that governs gene expressions in cells. The gene expression is controlled at different levels by many mechanisms, among which a key mechanism is mRNA transcription regulated by various proteins, known as transcription factors, which are bound to specific sites in the promoter region of a gene that activate or inhibit transcription. Using advanced molecular biology techniques, it has become possible to measure the gene expression levels (mRNA levels) of most genes in an organism simultaneously, hence making it possible to understand gene regulation and interactions.

In general, inference of a gene regulatory network is composed of three principal components: function modeling of the effect of a group of genes on a specific target gene, parameter estimation for function modeling of a specific network, and topology inference of regulatory network. As most genetic regulatory systems of interest involve many genes connected through interlocking positive and negative feedback loops, function modelings of interactions are important to unambiguously describe the structure of regulatory systems while predictions of their behavior can be made in a systematic way. Formal methods for the function modeling can be roughly categorized into qualitative models (such as graph models [1], Boolean function models [2, 3], and extended logical function models [4, 5]),

and quantitative models (such as the differential equations [6, 7], linear models [8, 9], nonlinear models [10, 11, 12], and radial basis function models [13]).

Typically, a specific network with a specific function model involves many parameters of biological interest and importance. In many cases these parameters are tuned in an ad hoc manner in an attempt to match experimental data [12]. Recent developments in simulation-based Bayesian inference techniques in principle allow direct inference of these parameters for any specified model from the data [14, 15]. For example, the Markov chain Monte Carlo (MCMC) method has been applied to obtain model parameters that are consistent with experimental data [13, 16].

Furthermore, revealing the structure of transcriptional regulation processes is a hard problem because of noisy data and incomplete information of regulation carried by data. Most analysis tools currently employed are based on clustering algorithms, which attempt to locate groups of genes that have similar expression patterns over a set of experiments. Unfortunately, such analysis is only useful in discovering genes that are coregulated. Recent effort on model system development has focused on Bayesian networks and Boolean networks, originally introduced in [14, 17], respectively. There are a number of works applying Boolean networks to genomic analysis, such as [2, 3, 17]. On the other hand, Bayesian learning techniques provide useful tools for inferring network structure based on experimental data while incorporating existing biological knowledge about gene interactions [14, 15].

This chapter gives a general overview of modeling of gene regulatory networks and discusses various statistical inference problems related to these models. Several review articles on modeling and simulation of gene regulatory systems exist in literature [18, 19, 20, 21]. This chapter differs from these articles in that it focuses on mathematical modeling issues of network structure, rather than on biological issues.

The remainder of this chapter is organized as follows. In Section 9.2 we review some existing gene function models. In Section 9.3, we discuss parameter estimation methods for various models. We then present network topology inference techniques in Section 9.4. Section 9.5 contains the conclusions.

9.2. Gene function modeling

9.2.1. Qualitative models

9.2.1.1. Directed and undirected graphs

A simple way to model a genetic regulation is to view it as a directed graph [1]. The interaction between a gene, say i, and a group of regulating genes J can be defined as $\langle i, J, S\rangle$, where S is a corresponding list of signs s indicating their regulatory influence, either activation ($s =$ "+") or inhibition ($s =$ "−"). Many databases contain such information about regulatory interactions. GeneNet [1], for example, contains descriptions of genes with their regulatory features, proteins and

protein complexes, regulatory interactions, and other chemical reactions for different types of cells. The databases and knowledge bases are usually supplemented by application to compose and edit networks by selecting and manipulating individual interactions. A number of operations on graphs can be carried out to make biological predictions about regulatory system.

9.2.1.2. Boolean networks

In Boolean networks [2, 3], each gene is described by a Boolean variable expressing that it is active or inactive and the dynamics describe how groups of genes act to change one another's states over time. Specifically, the state of a gene, say i, at time instant $t + \Delta t$ is determined by means of a Boolean function from the state of a group of genes, say k, at the previous time instant t. That is, $x_i(t + \Delta t) = b_i(\mathbf{x}_i(t))$, where b_i is a Boolean function with k inputs and $\mathbf{x}_i$ denotes the states of the k genes that regulate gene i. Such models are easy to implement, simplifying the examination of large sets of genes. A disadvantage of such a Boolean approach is that the abstraction of gene states to on/off makes it difficult to include many biological details of cellular mechanisms.

9.2.1.3. Generalized logical networks

We can also model the state of a gene by more than two values and allow transitions between states to occur asynchronously, then the Boolean network is generalized to a general discrete network [4, 5]. More precisely, the discrete variable X_i for gene i is an indication of the real concentration level. The value of X_i is defined by comparing the real concentration level of gene i with some thresholds. For example, gene i may have p possible values $x_i = m$, $m \in \{1, \ldots, p\}$ if it influences p other elements of the regulatory system and its value m is defined as $\delta_i^{(m)} < x_i \leq \delta_i^{(m+1)}$, where $\delta_i^{(m)}$ and $\delta_i^{(m+1)}$ are the threshold values. As a result, the logical function is a generalization of Boolean function since the logical values can now have more than two possible values. The choice of the logical function is made by biological considerations or a guess reflecting uncertainty about the structure of the system being studied.

9.2.2. Quantitative models

9.2.2.1. Differential equations

Ordinary differential equations have been widely used to analyze genetic relations between the concentration variables [6, 7]. More precisely, gene regulation is modeled by rate equations expressing the rate of production of a component of the system as a function of the concentrations of other components. Rate equations have the mathematical form

$$\frac{dx_i}{dt} = g(\mathbf{x}) - \gamma x_i, \tag{9.1}$$

where x_i denotes the cellular concentration of gene i and $\mathbf{x}$ is the vector of concentrations of proteins, mRNA, or small molecules; $g(\cdot)$ is a linear or nonlinear function and $\gamma > 0$ is the degradation rate of x_i. On the other hand, regulation of degradation could be modelled by replacing γ with a function similar to g. Discrete time delays, $\tau_{i,1}, \ldots, \tau_{i,n} > 0$, arising from the time required to complete transcription, translation, and diffusion to the place of action of a protein, can also be included in (9.1) to obtain

$$\frac{dx_i}{dt} = g\left(x_1(t-\tau_{i,1}), \ldots, x_n(t-\tau_{i,n})\right) - \gamma x_i, \quad 1 \le i \le n. \tag{9.2}$$

Such rate equations have been developed in the past century, in particular in the context of metabolic processes. Using these methods, kinetic models of genetic regulation processes can be constructed by specifying the function g.

A regular function often found in the literature is the so-called Hill curve given by

$$g^+(x_j, \theta_j, m) = \frac{x_j^m}{x_j^m + \theta_j^m}, \tag{9.3}$$

where θ_j is the threshold for the regulatory influence of x_j on a target gene, and $m > 0$ is a steepness parameter. The function takes values from 0 to 1 and increases with x_j, so that an increase in x_j will tend to increase the expression rate of the gene. In order to express that an increase in x_j decreases the expression rate, the regulation function can be replaced by $g^-(x_j, \theta_j, m) = 1 - g^+(x_j, \theta_j, m)$.

Due to the nonlinearity of g in (9.3), analytical solution of the rate equation is normally not possible. Therefore, a piecewise-linear differential equation model is often considered. In its most general form, the function g are replaced by

$$g_i(x) = \sum_{l \in L} \kappa_{i,l} \prod_{j=1} s^+\left(x_{l(j)}, \theta_{l(j)}\right)^{b_{i,l(j)}} \left[1 - s^+\left(x_{l(j)}, \theta_{l(j)}\right)\right]^{1-b_{i,l(j)}}, \tag{9.4}$$

where $\kappa_{i,l} > 0$ is a rate parameter; L is the possible sets of indices; $l(j)$ is the jth element in lth set of L; $b_{i,l(j)}$ takes values from $\{0, 1\}$; and s^+ is step function given by

$$s^+(x_j, \theta_j) = \begin{cases} 1, & x_j > \theta_j, \\ 0, & x_j < \theta_j. \end{cases} \tag{9.5}$$

9.2.2.2. Linear function models

In linear models [8, 9], the regulatory interactions take the form of linear functions. Let $x_i(t)$ denote the expression level of gene i at time t. Then the expression level of gene i at time $t + \Delta t$ is modelled by a basic linear model

$$x_i(t + \Delta t) = \sum_j w_{i,j} x_j(t) + \sum_k v_{i,k} \mu_k(t) + b_i, \tag{9.6}$$

where $w_{i,j}$ and $v_{i,k}$ indicate how much the level of gene j and the kth input $\mu_k(t)$ influence gene i; and b_i is the basal expression level of gene i. That is, the influence of all genes are summarized in a linear gene-to-gene relationship. These weights provide information about the relationships between genes, that is, zero weights indicate the absence of interaction and a positive or negative weight corresponds to stimulation or repression. The absolute value of a weight corresponds to the strength of the interaction. Of course, a linear model can never be much more than a caricature of the real system. Nevertheless, the linear model often performs surprisingly well compared with other complex models.

9.2.2.3. Nonlinear function models

The nonlinear function models directly characterize effects that result from the combination of the gene expression levels [10, 11, 12]. The influence of all genes are also summarized in the linear gene-to-gene relationship whereas these weights provide information about the relationships between genes. The absolute value of a weight corresponds to the strength of the interaction. Then the influence of groups of genes on one gene can be represented by the following generalized differential equation

$$\frac{dx_i(t)}{dt} = r_i g\left(\sum_j w_{i,j}x_j(t) + \sum_k v_{i,k}\mu_k(t) + b_i\right) - \lambda_i x_i(t), \tag{9.7}$$

or difference equation of the similar form

$$x_i(t+\Delta t) = r_i g\left(\sum_j w_{i,j}x_j(t) + \sum_k v_{i,k}\mu_k(t) + b_i\right) - \lambda_i x_i(t), \tag{9.8}$$

where $x_i(t)$ is gene expression of gene i at time instant t; $w_{i,j}$ and $v_{i,k}$ indicate how much the level of gene j and the kth input $\mu_k(t)$ influence gene i; b_i is the basal expression level of gene i; and λ_i is the degradation constant of the ith gene expression product; g is the monotonic regulation expression function, and often takes the form of

$$g(x) = \frac{1}{1+\exp\left(-\alpha_i x - \beta_i\right)}, \tag{9.9}$$

where α_i and β_i are two specific constants that define the shape of the dose-response curve of gene i. This assumes that each gene has a static dose-dependent response activating and repressing regulatory influences. The constant α_i can be any positive real number and defines the slope of the curve at its inflection point. Whereas the constant β_i defines the curve's vertical intercept, where the positive and negative regulatory inputs are equal. This point corresponds conceptually to the gene basal transcription. Positive β_i represents genes with high basal levels of transcription where negative β_i represents genes with low basal levels of transcription. Note that if $g(x) = x$, then the nonlinear model is simplified to a linear model.

9.2.2.4. Radial basis function

The radial basis function can be employed to characterize any multiple-input-single-output relationships between genes [13]:

$$x_i(t+\Delta t) = \sum_j a_j \phi(||\mathbf{x}(t) - \boldsymbol{\mu}_j||) + b + \boldsymbol{\beta}^T \mathbf{x}(t), \tag{9.10}$$

where $\mathbf{x}(t) = [x_1(t), \ldots, x_n(t)]^T$; ϕ is a radial basis function (RBF); $\|\cdot\|$ denotes a distance metric (usually Euclidean or Mahalanobis); $\boldsymbol{\mu}_j$ denotes the jth RBF center; a_j is the jth RBF coefficient; b is the basal level; and $\boldsymbol{\beta}$ is the linear regression parameter vector. Depending on the a priori knowledge about the smoothness of mapping, we can choose different types of basis functions, for example, the Gaussian basis function $\phi(\rho) = \exp(-\rho^2)$. Since the RBF can approximate any function, it can be used to characterize gene function modeling without much prior biological knowledge. This is especially true for the function modeling between clustered genes.

9.3. Parameter estimation with known network structure

One goal of regulation network modeling is to predict the genetic pathways that underlie observed gene expression data. Given a certain function model for gene interactions, the first step is to estimate the parameters within the model, based on experimental data. In this case, we assume that the parameter values are constant across time. Then given only input/output data sets, we want to identify parameter values that define the regulatory network. The hope is that if our modeling scheme is a reasonable approximation of the true regulatory network, we may use it to predict genetic pathways from experimentally derived expression data.

9.3.1. Equation-based methods

Equation-based algorithms relate the expression of each gene to the expression level of all other genes in the form of equations [7, 12]. These equations can be linear, nonlinear, and/or differential equations. Putative regulators are identified by solving the set of equations for the parameters which relate each gene to the other genes. These parameters represent the regulatory influence of each gene on the others. Specifically, the algorithms employed to infer model parameters from measured mRNA levels often require solving a least-square system of linear equations, or implementing a nonlinear optimization procedure for nonlinear modeling.

Here, we consider a special case (without degradation, i.e., $\lambda_i = 0$) of nonlinear model (9.7) or (9.8) to show how to perform linear regression to estimate parameters based on the experimental data. For simplicity, we also omit the input

influence, that is,

$$x_i(t+\Delta t) = r_i g\left(\sum_j w_{i,j} x_j(t)\right), \tag{9.11}$$

where $g(\cdot)$ takes the form of (9.9). We can rewrite the above formulation as

$$s_i(t+\Delta t) = \sum_j W_{i,j} x_j(t) + \beta_i, \tag{9.12}$$

$$x_i(t+\Delta t) = \frac{1}{1+\exp\left(-s_i(t+\Delta t)\right)}, \tag{9.13}$$

where $W_{i,j} = \alpha_i w_{i,j}$ with α_i and β_i defined in (9.9). After "desquashing" the relative expression by $s_i(t) = -\ln(1/x_i(t) - 1)$, we rewrite the model (9.12) in vector form as

$$\underbrace{\begin{bmatrix} s_i(\Delta t) \\ s_i(2\Delta t) \\ \vdots \\ s_i(T\Delta t) \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} x_1(0) & x_2(0) & \cdots & x_J(0) & 1 \\ x_1(\Delta t) & x_2(\Delta t) & \cdots & x_J(\Delta t) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1(T\Delta t - \Delta t) & x_2(T\Delta t - \Delta t) & \cdots & x_J(T\Delta t - \Delta t) & 1 \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} W_{i,1} \\ W_{i,2} \\ \vdots \\ W_{i,J} \\ \beta_i \end{bmatrix}}_{\mathbf{a}}. \tag{9.14}$$

Therefore, given the measurement data, we can then construct the matrix **M** and the vector **b** corresponding to the desquashing relative expression level of the gene of interest, and estimate the unknown vector **a** corresponding to the gene of interest. If the linear system equation is overdetermined, that is, there are more observations than the number of genes, then we can use the least-square solution for **a**. However, if there are a fewer observations than the number of genes, the estimation problem is "underdetermined" and there are many equally good solutions to **a**. We resort to some prior knowledge about the network, to choose an optimal solution.

On the other hand, if we need to consider the degradation, that is, $\lambda_i \neq 0$, in the nonlinear model (9.7) or (9.8), more complex optimization algorithms, such as the simulated annealing algorithm [22] or the genetic algorithm [23] can be employed to obtained the parameters for the nonlinear model. The basic idea of these algorithms is as follows. We first need to define a suitable objective function, for example, the fitness error [22, 23]

$$f = \frac{1}{1 + \left(1/(T-1)\right) \sum_t \left(x_i(t) - x_i^d(t)\right)^2}, \tag{9.15}$$

where $x_i(t)$ and $x_i^d(t)$ indicate the observed expression level for gene i at time t and the corresponding expression level obtained from the model (9.7) or (9.8)

with given parameters. Then the values of these model parameters are tuned using either the simulated annealing algorithm or the generic algorithm to attain the global extremum of the goal function.

The essential problem with using the equation-based methods is that the number of connection and strength parameters grows quadratically in the number of genes. Some form of dimensionality reduction is needed. Fortunately dimensionality reduction is available in the present practice of clustering the large-scale time course expression data by genes, into gene clusters [9, 24, 25]. In this way one can derive a small number of cluster-mean time courses for the "aggregated genes." Then the equation-based methods can be employed on the aggregated genes to characterize the regulatory network.

9.3.2. Bayesian methods

One of the most commonly used Bayesian inference algorithms is the junction tree algorithm [26] which infers Bayesian networks with discrete and continuous variables, and provides efficient techniques for handling networks with many observed nodes. However, exact inference in densely connected Bayesian networks is often computationally intractable, so we must resort to approximates [27]. We illustrate the Bayesian method, for example, by a gene selection method for the multiclass cancer classification using multinomial probit regression model [13].

Assume there are K classes of cancers. Let $\mathbf{w} = [w_1, \ldots, w_m]^T$ denote the class labels, where $w_i = k$ indicates the sample i being cancer k, where $k = 1, \ldots, K$. Assume there are n genes. Let x_{ij} be the measurement of the expression level of the jth gene for the ith sample where $j = 1, 2, \ldots, n$. Let $\mathbf{X} = (x_{ij})_{m,n}$ denote the expression levels of all genes, that is,

$$\mathbf{X} = \begin{bmatrix} \text{gene } 1 & \text{gene } 2 & \cdots & \text{gene } n \\ x_{11} & x_{12} & \cdots & x_{1n} \\ x_{11} & x_{12} & \cdots & x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}. \tag{9.16}$$

Let $\mathbf{X}_i$ denote the ith row of matrix $\mathbf{X}$. In the binomial probit regression, that is, $K = 2$, the relationship between w_i and the gene expression levels $\mathbf{X}_i$ is modeled by using a probit regression model [28], a special linear model (9.6), which yields

$$P(w_i = 1 \mid \mathbf{X}_i) = \Phi(\mathbf{X}_i \boldsymbol{\beta}), \quad i = 1, \ldots, m, \tag{9.17}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)^T$ is the vector of regression parameters and Φ is the standard normal cumulative distribution function. Introduce m-independent latent variable $\mathbf{z} = [z_1, \ldots, z_m]^T$, where $z_i \sim \mathcal{N}(\mathbf{X}_i \boldsymbol{\beta}, 1)$, that is,

$$z_i = \mathbf{X}_i \boldsymbol{\beta} + e_i, \quad i = 1, \ldots, m, \tag{9.18}$$

and $e_i \sim \mathcal{N}(0,1)$. Define $\boldsymbol{\gamma}$ as the $n \times 1$ indicator vector with the jth element γ_j such that $\gamma_j = 0$ if $\beta_j = 0$ (the variable is not selected) and $\gamma_j = 1$ if $\beta_j \neq 0$ (the variable is selected). The Bayesian variable selection is to estimate $\boldsymbol{\gamma}$ from the posterior distribution $p(\boldsymbol{\gamma} \mid \mathbf{z})$.

However, when $K > 2$, the situation is different from the binomial case because we have to construct $K-1$ regression equations similar to (9.18). Introduce $K-1$ latent variables $y_1, \ldots, y_{K-1}$ and $K-1$ regression equations such that $y_k = \mathbf{X}\boldsymbol{\beta}_k + e_k$, $k = 1, \ldots, K-1$, where $e_k \sim \mathcal{N}(0,1)$. Let y_k take m values $\{y_{k,1}, \ldots, y_{k,m}\}$ for each equation. In matrix form,

$$\begin{aligned} y_{1,i} &= \mathbf{X}_i\boldsymbol{\beta}_1 + e_{1,i}, \\ \vdots \quad &\vdots \quad \vdots \qquad i = 1, \ldots, m. \\ y_{K-1,i} &= \mathbf{X}_i\boldsymbol{\beta}_{K-1} + e_{K-1,i}, \end{aligned} \tag{9.19}$$

Denote $\mathbf{y}_k \triangleq [y_{k,1}, \ldots, y_{k,m}]^T$ and $\mathbf{e}_k \triangleq [e_{k,1}, \ldots, e_{k,m}]^T$. Then (9.19) can be rewritten as

$$\mathbf{y}_k = \mathbf{X}\boldsymbol{\beta}_k + \mathbf{e}_k, \quad k = 1, \ldots, K-1. \tag{9.20}$$

This model is called multinomial probit model, which is a special form of linear model. For background on multinomial probit models, see [29]. Note that we do not have the observations of $\{\mathbf{y}_k\}_{k=1}^{K-1}$, which makes it difficult to estimate the parameters in (9.20).

We consider gene selection scheme to select the different strongest genes for each equation in (9.20). Given $\boldsymbol{\gamma}_k$, let $\boldsymbol{\beta}_{\boldsymbol{\gamma}_k}$ consist of all nonzero elements of $\boldsymbol{\beta}$ and let $\mathbf{X}_{\boldsymbol{\gamma}_k}$ be the columns of $\mathbf{X}$ corresponding to those of $\boldsymbol{\gamma}$ that are equal to 1 for equation k. Then (9.20) is rewritten as

$$\mathbf{y}_k = \mathbf{X}_{\boldsymbol{\gamma}_k}\boldsymbol{\beta}_{\boldsymbol{\gamma}_k} + \mathbf{e}_k, \quad k = 1, \ldots, K-1. \tag{9.21}$$

Now the problem is how to estimate $\boldsymbol{\gamma}_k$ and the corresponding $\boldsymbol{\beta}_k$ and $\mathbf{y}_k$ for each equation in (9.21).

A Gibbs sampler [30, 31] is employed to estimate all the parameters. Given $\boldsymbol{\gamma}_k$ for equation k, the prior distribution of $\boldsymbol{\beta}_{\boldsymbol{\gamma}_k}$ is $\boldsymbol{\beta}_{\boldsymbol{\gamma}_k} \sim \mathcal{N}(0, c(\mathbf{X}_{\boldsymbol{\gamma}_k}^T\mathbf{X}_{\boldsymbol{\gamma}_k})^{-1})$ [32], where c is a constant. The detailed derivation of the posterior distributions of the parameters are same as that in [32, 33]. Here we summarize the procedure for Bayesian variable selection. Denote

$$S(\boldsymbol{\gamma}_k, \mathbf{y}_k) = \mathbf{y}_k^T\mathbf{y}_k - \frac{c}{c+1}\mathbf{y}_k^T\mathbf{X}_{\boldsymbol{\gamma}_k}(\mathbf{X}_{\boldsymbol{\gamma}_k}^T\mathbf{X}_{\boldsymbol{\gamma}_k})^{-1}\mathbf{X}_{\boldsymbol{\gamma}_k}^T\mathbf{y}_k, \quad k = 1, \ldots, K-1. \tag{9.22}$$

Then the Gibbs sampling algorithm for estimating $\{\boldsymbol{\gamma}_k, \boldsymbol{\beta}_k, \mathbf{y}_k\}$ is as follows.

(i) Draw $\boldsymbol{\gamma}_k$ from $p(\boldsymbol{\gamma}_k \mid \mathbf{y}_k)$, where

$$p(\boldsymbol{\gamma}_k \mid \mathbf{y}_k) \propto (1+c)^{-n_{\gamma_k}/2} \exp\left[-\frac{1}{2}S(\boldsymbol{\gamma}_k, \mathbf{y}_k)\right] \prod_{j=1}^{n} \pi_j^{\gamma_{k,j}} (1-\pi_j)^{1-\gamma_{k,j}}, \tag{9.23}$$

where $n_{\gamma_k} = \sum_{j=1}^{n} \gamma_{k,j}$ and $\pi_j = P(\gamma_{k,j} = 1)$ is the prior probability to select the jth gene. We sample each $\gamma_{k,j}$ independently from

$$p(\gamma_{k,j} \mid \mathbf{y}_k, \gamma_{k,i\neq j}) \propto (1+c)^{-n_{\gamma_k}/2} \exp\left[-\frac{1}{2} S(\boldsymbol{\gamma}_k, \mathbf{y}_k)\right] \pi_j^{\gamma_{k,j}} (1-\pi_j)^{1-\gamma_{k,j}}, \quad j = 1, \ldots, n. \tag{9.24}$$

(ii) Draw $\boldsymbol{\beta}_k$ from

$$p(\boldsymbol{\beta}_k \mid \boldsymbol{\gamma}_k, \mathbf{y}_k) \propto \mathcal{N}(\mathbf{V}_{\boldsymbol{\gamma}_k} \mathbf{X}_{\boldsymbol{\gamma}_k}^T \mathbf{y}_k, \mathbf{V}_{\boldsymbol{\gamma}_k}), \quad \text{where } \mathbf{V}_{\boldsymbol{\gamma}_k} = \frac{c}{1+c} (\mathbf{X}_{\boldsymbol{\gamma}_k}^T \mathbf{X}_{\boldsymbol{\gamma}_k}^{-1}). \tag{9.25}$$

(iii) Draw $\mathbf{y}_k = [y_{k,1}, \ldots, y_{k,m}]$, $k = 1, \ldots, K$ from a truncated normal distribution as follows [34].

For $i = 1, 2, \ldots, m$,

(a) if $w_i = k$, then draw $y_{k,i}$ according to $y_{k,i} \sim \mathcal{N}(\mathbf{X}_{\gamma_k} \boldsymbol{\beta}_k, 1)$ truncated left by $\max_{j\neq k} y_{j,i}$, that is,

$$y_{k,i} \sim \mathcal{N}(\mathbf{X}_{\gamma_k} \boldsymbol{\beta}_k, 1) 1_{\{y_{k,i} > \max_{j\neq k} y_{j,i}\}}; \tag{9.26}$$

(b) else $w_i \neq j$ and $j \neq k$, then draw $y_{j,i}$ according to $y_{j,i} \sim \mathcal{N}(\mathbf{X}_{\gamma_j} \boldsymbol{\beta}_j, 1)$ truncated right by the newly generated $y_{k,i}$, that is,

$$y_{j,i} \sim \mathcal{N}(\mathbf{X}_{\gamma_j} \boldsymbol{\beta}_j, 1) 1_{\{y_{j,i} \leq y_{k,i}\}}. \tag{9.27}$$

Endfor.

9.4. Inference for network topology

Inference of gene regulatory network structure has appeared in the past few years as a method to deal with a large amount of gene expression data available from measurements. Using expression level information from either multiple samples of a system in different states or a time series of points, these algorithms calculate which genes appear to be regulators of other genes, that is, which genes increase or decrease the expression of other genes.

There are two major types of network topology inference algorithms: pairwise and network-based. The pairwise methods consist of finding pairs of genes whose expression levels are correlated, and suggesting one to be the putative regulator of another. Because these methods have no actual model that describes exactly how genes are activated by external inputs and other genes, no prediction of gene expression can be made.

Network-based algorithms come in two basic types: Boolean and Bayesian. Boolean networks assume genes have only two states, on and off and genes are

connected to each other with logical relationships. Therefore, for Boolean network, it is easy to figure out what logical rule each node is using, for example, by exhaustively enumerating them and finding the one that fits data. However, Bayesian networks represent probabilistic connections between genes. A regulatory link between two genes indicates that knowing the value of one helps predict the value of the other. In principle, the methods outlined in Section 9.3 can be extended to incorporate uncertainty regarding network topology. The full space of possible topology can be explored using reverse-jump MCMC techniques. More precisely, these procedures introduce an additional step into the MCMC algorithm whereby small changes to the network topology are proposed. These changes are accepted with a carefully constructed probability which ensures that the resulting equilibrium distribution is the posterior distribution of interest.

9.4.1. Pairwise methods

Pairwise methods construct relationships between genes based solely on pairwise comparisons [6, 35, 36, 37]. Therefore they do not take into account interactions where the resulting expression level of one gene is governed by the combined action of multiple other genes. Common methodologies include smoothing data, extracting trends from data, labelling clustered data, categorizing data, and representing suitably analyzed data in suggestive visual forms [35]. Extensions to this basic idea include identifying common cis-acting sequence motifs within clusters [38] and correlating lagged data vectors from time series data [6]. This paradigm was proven quite successful in identifying a number of striking patterns within gene expression data. For example, various genes of similar function often cluster together, especially when the topological clusters are optimally ordered, and various genes that have been identified, seem to offer predictive power in categorizing types of cancers.

For instance, it is proposed in [6] to express regulation based on whether peaks in one signal precede peaks in another signal. After thresholding and clustering, each prototype is represented as a series of peaks, resulting in a set of prototype signals. For each pair of prototypes three scores are computed, representing a possible activating, inhibiting, or unmatching relationship. The regulation matrix is inferred by taking for each pair of genes the highest of these three scores.

Another well-known method is the correlation metric method, which first computes the magnitude and position at which the maximal cross-correlation occurs [35]. This provides measures of similarity and temporal ordering, respectively. Then a distance matrix is constructed by comparing for each pair of genes, their similarities to other genes. The significant eigenvalues of the constructed distance matrix provides an indication of the intrinsic dimensionality of the system. Single-linkage hierarchical clustering is employed to find a singly linked tree that connects associated genes. This tree is augmented with directional and time-lag information, revealing temporal ordering.

9.4.2. Bayesian methods

Very recently, a full Bayesian approach to infer the gene regulatory networks from expression data was developed in [39]. In what follows, we briefly review this technique.

9.4.2.1. Bayesian score

Consider a finite set of $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ of discrete random variables where each variable X_i may take on values from a finite set. A Bayesian network is an annotated directed acyclic graph that encodes a joint probability distribution over $\mathcal{X}$. More precisely, a Bayesian network for $\mathcal{X}$ is a pair $B = \langle G, \theta \rangle$, where the first component G is a directed acyclic graph whose vertices correspond to the genes, $X_1, X_2, \ldots, X_n$ and the second component θ represents probability distribution for each node: $\theta_i = \theta_{u_i|pa(x_i)}$ for each possible value u_i of x_i conditioned on one instance $pa(x_i)$ of the set of parents $Pa(X_i)$. If more than one graph is discussed, then we use $Pa^G(X_i)$ to specify X_i's parents in graph G. A Bayesian network B specifies a unique joint probability distribution over $\mathcal{X}$ given by

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid Pa^G(x_i)). \tag{9.28}$$

The problem of learning a Bayesian gene regulatory network can be stated as follows: given a set of gene expression measurements $\mathcal{O} = \{\underline{x}_1, \ldots, \underline{x}_m\}$, where $\underline{x}_i = [x_{i1}, \ldots, x_{in}]^T$ is the observation vector of node x_i, find a network B that best matches $\mathcal{O}$. The common approach to learn Bayesian networks is to search for the networks with the highest a posteriori probabilities

$$P(G \mid \mathcal{O}) \propto P(\mathcal{O} \mid G)P(G), \tag{9.29}$$

where $P(G)$, the prior probability for the network G, is assumed to take a biological knowledge about the network or a uniform distribution on all possible topologies and $P(\mathcal{O} \mid G)$ is called Bayesian score. Note that

$$P(\mathcal{O} \mid G) = \int p(\mathcal{O} \mid G, \Theta) p(\Theta)\, d\Theta. \tag{9.30}$$

If the analytical form for the integration can be found, the Bayesian score can be easily implemented. However, the computation in (9.30) is often prohibitive; thus instead of integration, the Bayesian score is approximated with the value at estimated parameters $\hat{\Theta}$, that is,

$$P(\mathcal{O} \mid G) \approx p(\mathcal{O} \mid G, \hat{\Theta}) p(\hat{\Theta}). \tag{9.31}$$

We also impose a conditional independence assumption on $p(\mathcal{O} \mid G, \Theta)$:

$$p(\mathcal{O} \mid G, \Theta) \triangleq p(\underline{x}_1, \ldots, \underline{x}_n \mid G, \Theta) = \prod_{i=1}^{n} p(\underline{x}_i \mid Pa(x_i), \Theta_i),$$
$$p(\Theta) \triangleq p(\Theta^{(1)}, \ldots, \Theta^{(n)}) = \prod_{i=1}^{n} p(\Theta^{(i)}), \tag{9.32}$$

where $p(\Theta^{(i)})$ is a prior density for the parameters of gene function modeling and $pa(x_i)$ denotes the observations corresponding to the parent nodes of x_i in G. Assuming node i has $v(i)$ parent nodes, $pa(x_i) = [\underline{x}_{i_1}, \underline{x}_{i_2}, \ldots, \underline{x}_{i_{v(i)}}]$.

Specifically, we assume that the function modeling of gene i is modelled by the RBF (9.10) with independent Gaussian noise. Let J_i and d_i be the number of RBFs and regression parameters in (9.10) for gene i, respectively. Denote

$$\boldsymbol{\alpha}_i = [b_i, \beta_{i,1}, \beta_{i,2}, \ldots, \beta_{i,d_i}, a_{i,1}, \ldots, a_{i,J_i}],$$
$$\mathbf{D}_i = \begin{bmatrix} 1 & x_{i,1,1} \cdots x_{i,1,d_i} & \phi(||\underline{x}_{i,1} - \boldsymbol{\mu}_{i,1}||) \cdots \phi(||\underline{x}_{i,1} - \boldsymbol{\mu}_{i,J_i}||) \\ 1 & x_{i,2,1} \cdots x_{i,2,d_i} & \phi(||\underline{x}_{i,2} - \boldsymbol{\mu}_{i,1}||) \cdots \phi(||\underline{x}_{i,2} - \boldsymbol{\mu}_{i,J_i}||) \\ \vdots & \vdots & \vdots \\ 1 & x_{i,v(i),1} \cdots x_{i,v(i),d_i} & \phi(||\underline{x}_{i,v(i)} - \boldsymbol{\mu}_{i,1}||) \cdots \phi(||\underline{x}_{i,v(i)} - \boldsymbol{\mu}_{i,J_i}||) \end{bmatrix}, \tag{9.33}$$

then we have the regulatory model for gene i in a vector-matrix form

$$\underline{x}_i = \mathbf{D}_i \boldsymbol{\alpha}_i + \mathbf{n}_i, \tag{9.34}$$

where $\mathbf{n}_i$ is the vector of measurement noise at different time instants for gene i. Therefore, the likelihood function $p(\underline{x}_i \mid Pa(x_i), \Theta^{(i)})$ is easily written as

$$p(\underline{x}_i \mid Pa(x_i), \Theta^{(i)}) = (2\pi\eta_i)^{-v(i)/2} \exp\left(-\frac{1}{2\eta_i} ||\underline{x}_i - \mathbf{D}_i \boldsymbol{\alpha}_i||^2 \right). \tag{9.35}$$

Given $\{J_i, \boldsymbol{\mu}_{i,1}, \ldots, \boldsymbol{\mu}_{i,J_i}\}$ and d_i, the least-squares estimate of $\boldsymbol{\alpha}$ is given by

$$\hat{\boldsymbol{\alpha}}_i = (\mathbf{D}_i^T \mathbf{D}_i)^{-1} \mathbf{D}_i^T \underline{x}_i, \tag{9.36}$$

and the estimation of η_i is given by

$$\hat{\eta}_i = \frac{1}{v(i)} (\underline{x}_i - \mathbf{D}_i \hat{\boldsymbol{\alpha}}_i)^T (\underline{x}_i - \mathbf{D}_i \hat{\boldsymbol{\alpha}}_i) = \frac{1}{v(i)} \underline{x}_i^T \mathbf{P}_i^* \underline{x}_i, \tag{9.37}$$

where

$$\mathbf{P}_i^* \triangleq \mathbf{I}_{v(i)} - \mathbf{D}_i (\mathbf{D}_i^T \mathbf{D}_i)^{-1} \mathbf{D}_i^T. \tag{9.38}$$

Based on the minimum description length (MDL) criterion, we can impose the following a priori distribution on J_i and d_i [40]:

$$P(J_i, d_i) \propto \exp\left[-\left(J_i + \frac{d_i + 1}{2}\right) \log \nu(i) \right]. \tag{9.39}$$

Assuming the noise samples are i.i.d. Gaussian, it can then be shown that the joint posterior distribution of $(J_i, \boldsymbol{\mu}_{i,1}, \ldots, \boldsymbol{\mu}_{i,J_i})$ is given by [40],

$$p(J_i, \boldsymbol{\mu}_{i,1}, \ldots, \boldsymbol{\mu}_{i,J_i} \mid \mathcal{O}) \propto \left[(\underline{x}_i^T \mathbf{P}_i^* \underline{x}_i)^{-\nu(i)/2} \right] P(J_i, d_i). \tag{9.40}$$

Hence the maximum a posteriori (MAP) estimate of these parameters is obtained by maximizing the right-hand side of (9.40) or by sampling algorithm such as MCMC algorithm.

Based on estimated parameters $\hat{\Theta}_i$, the Bayesian score $P(\mathcal{O} \mid G)$ is approximated by

$$\begin{aligned} P(\mathcal{O} \mid G) &\approx \prod_{i=1}^{n} p(\underline{x}_i \mid Pa(x_i), \hat{\Theta}_i)\, p(\hat{\Theta}_i) \\ &\propto \prod_{i=1}^{n} (\underline{x}_i^T \mathbf{P}_{J_i}^* \underline{x}_i)^{-\hat{\nu}(i)/2} \exp\left[-\left(\hat{J}_i + \frac{\hat{d}_i + 1}{2}\right) \right]. \end{aligned} \tag{9.41}$$

The computation of Bayesian score for other function modeling can be obtained in a similar way. Once the Bayesian score is calculated, standard heuristic search techniques, such as greedy hill-climbing and simulated annealing algorithm, can be employed to find the network with the highest scores. Such search procedures do not need any prior knowledge network structure. For example, the greedy hill-climbing search starts with some seeds, say some three genes with two high scoring edges. Then add or remove a gene at each step. Once it reaches a local minimum, it repeats the procedure until all seeds are used. Finally the subnetworks with highest scores are obtained.

9.4.2.2. Searching network via MCMC

Within the Bayesian framework, a fully Bayesian approach to constructing regulation network is introduced in [13] by searching over the space of all possible network topologies and picking those with the highest Bayesian scores. Given a network configuration G, we calculate the parameters Θ associated with it as well as the corresponding Bayesian score $P(\mathcal{O} \mid G)$. We then set the network configuration by implementing a reversible jump MCMC step. The process repeats a sufficient number of iterations. Finally, the regulation network is formed by choosing those networks with the highest Bayesian scores. More precisely, we generate an initial directed graph $G^{(0)}$, say, by clustering, and compute $P(\mathcal{O} \mid G)$; and then, for $j = 1, 2, \ldots$, we compute the Bayesian score $P(\mathcal{O} \mid G)$, and pick $G^{(j+1)}$ via an MCMC step.

The space of all possible network topologies is huge. To find the networks with highest scores, we resort to the MCMC strategy. Given the current network topology G, define its neighborhood, $\aleph(G)$, to be the set of graphs which differ by one edge from G, that is, we can generate $\aleph(G)$ by considering all single-edge additions, deletions, and reversals [16]. Let $q(G' \mid G) = 1/|\aleph(G)|$, for $G' \in \aleph(G)$, and $q(G' \mid G) = 0$ for $G' \notin \aleph(G)$. Sample G' from G by a random single-edge addition, deletion, or reversal in G. Then the acceptance ratio is given by

$$R = \frac{q(G \mid G')P(G' \mid \mathcal{O})}{q(G' \mid G)P(G \mid \mathcal{O})} = \frac{P(\mathcal{O} \mid G')}{P(\mathcal{O} \mid G)}, \tag{9.42}$$

where $P(\mathcal{O} \mid G)$ and $P(\mathcal{O} \mid G')$ can be obtained from (9.41).

We pick the networks with the highest Bayesian scores (although this could be modified by the prior knowledge). After selecting the K graphs $\{\mathcal{N}_k\}_{k=1}^{K}$ with the highest scores out of a large number of networks generated by the MCMC technique, we next estimate the probability $P(x \to y)$ from node x to node y using Monte Carlo methods. For example, the Markov chain length is chosen as n_1 after the n_2 burn-in period, then the probability $p(x \to y)$ is estimated by

$$P(x \to y) \approx \sum_{j=n_1}^{n_1+n_2} \delta(x, y, G_j), \tag{9.43}$$

where $\delta(x, y, G_j)$ is 1 if G_j contains the link $x \to y$ and zero otherwise. Thus we can compute the posterior probability of all possible edges, and then the following method can be used to construct gene regulatory networks.

(i) Select a significant confidence α and then construct a graph over variables with an edge between x and y if this Markov pair is confident $P(x \to y) > \alpha$.
(ii) Take each nontrivial component as a seed of a subnetwork.
(iii) Expand the seed by adding variables that are related to this seed by a Markov pair with confidence level above another parameter $\alpha' < \alpha$.
(iv) Repeat the procedure and finally obtain the gene regulatory network.

9.4.2.3. Structure EM algorithm

We next model the temporal processes by a dynamic Bayesian network and then solve it using the structure EM algorithm [41]. In this case, we not only model a probability distribution over a fixed number of genes, but also the joint distribution over all possible trajectories of a process. For simplicity, we assume that the process is Markovian and stationary. In other words, the dynamic Bayesian network can be represented by

(i) a prior network B_0 that specifies a distribution over initial state $\mathbf{X}[0]$;
(ii) a transition network $\mathbf{B}_{\to}$ that is taken to specify the transition probability $P(\mathbf{X}_{t+1} \mid \mathbf{X}_t)$.

Here $\mathbf{X}[t]$ are the expression levels for genes at time t. Given a dynamic Bayesian network, the distribution can be represented by

$$P_B(\mathbf{x}[0],\ldots,\mathbf{x}[T]) = P_{B_0}(\mathbf{x}[0]) \prod_{t=0}^{T} P_{B_{\rightarrow}}(\mathbf{x}[t] \mid \mathbf{x}[t-1]). \tag{9.44}$$

From (9.44), we can compute the Bayesian score $P(B_0, B_{\rightarrow} \mid \mathbf{x}[0],\ldots,\mathbf{x}[T])$ using a decomposition method. Furthermore, a hill-climbing search procedure can be employed to improve a candidate structure by applying the best arc addition, deletion, or reversal [15, 41]. The structure EM algorithm can be efficiently used to learn the temporal structure and parameters. It iteratively alternates between evaluating the expected score of a model and changing the model structure, until a local maximum is reached. More precisely, the procedure of the structure EM algorithm for the inference of gene network can be described as follows [41].

(i) Choose $(B_0, B_{\rightarrow})$ from a prior distribution.
(ii) Improve the parameters of $(B_0, B_{\rightarrow})$ using EM.
(iii) Search the possible structure by a hill-climbing algorithm using the expected estimation computed with EM.
(iv) Set the best scoring structure as $(B_0^{n+1}, B_{\rightarrow}^{n+1})$.
(v) Stop the iteration if the new structure satisfies some stopping rules; otherwise, repeat from the second step.

One way to scale the structure EM algorithm for large gene networks might be to first perform a clustering of the time series of the observed variables and then to associate a transition structure with these clusters. The result would be a Markov model with a structured "backbone."

9.5. Concluding remarks

In this chapter, we have briefly reviewed the mathematical modeling issues for transcriptional regulatory networks. We summarized some existing qualitative and quantitative models for regulatory networks. We also discussed parameter estimation methods for these models, ranging from linear least-square methods to Bayesian inference methods. Finally, we discussed some recent development on inference of network topologies from experimental data. In particular, a full Bayesian framework using reverse-jump MCMC method is discussed. In summary, we have seen that advanced statistical inference techniques will play a vital role in future quantitative generic research.

Acknowledgment

This work was supported in part by the US National Science Foundation (NSF) under Grant NSF DMS 0244583.

Bibliography

[1] F. A. Kolpakov, E. A. Ananko, G. B. Kolesov, and N. A. Kolchanov, "GeneNet: a gene network database and its automated visualization," *Bioinformatics*, vol. 14, no. 6, pp. 529–537, 1998.

[2] L. Glass and S. A. Kauffman, "The logical analysis of continuous, non-linear biochemical control networks," *J. Theor. Biol.*, vol. 39, no. 1, pp. 103–129, 1973.

[3] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, NY, USA, 1993.

[4] R. Thomas, "Regulatory networks seen as asynchronous automata: a logical description," *J. Theor. Biol.*, vol. 153, pp. 1–23, 1991.

[5] R. Thomas, "Laws for the dynamics of regulatory networks," *Int. J. Dev. Biol.*, vol. 42, no. 3, pp. 479–485, 1998.

[6] T. Chen, V. Filkov, and S. Skiena, "Identifying gene regulatory networks from experimental data," in *Proc. 3rd Annual International Conference on Computational Molecular Biology*, pp. 94–103, Lyon, France, 1999.

[7] T. Mestl, E. Plahte, and S. W. Omholt, "A mathematical framework for describing and analyzing gene regulatory networks," *J. Theor. Biol.*, vol. 176, pp. 291–300, 1995.

[8] P. D'Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear modeling of mRNA expression levels during CNS development and injury," *Pac. Symp. Biocomput.*, vol. 4, pp. 41–52, 1999.

[9] E. P. van Someren, L. F. Wessels, and M. J. Reinders, "Linear modeling of genetic networks from experimental data," *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, vol. 8, pp. 355–366, 2000.

[10] E. Mjolsness, D. H. Sharp, and J. Reinitz, "A connectionist model of development," *J. Theor. Biol.*, vol. 152, no. 4, pp. 429–453, 1991.

[11] D. C. Weaver, C. T. Workman, and G. D. Stormo, "Modeling regulatory networks with weight matrices," *Pac. Symp. Biocomput.*, vol. 4, pp. 112–123, 1999.

[12] L. F. Wessels, E. P. van Someren, and M. J. Reinders, "A comparison of genetic network models," *Pac. Symp. Biocomput.*, vol. 6, pp. 508–519, 2001.

[13] X. Zhou, X. Wang, and E. R. Dougherty, "Gene prediction using multinomial probit regression with Bayesian gene selection," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 115–124, 2004.

[14] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian network to analyze expression data," *J. Comput. Biol.*, vol. 7, pp. 601–620, 2000.

[15] K. Murphy and S. Mian, "Modelling gene expression data using dynamic Bayesian networks," Tech. Rep., University of California, Berkeley, Calif, USA, 1999.

[16] P. Giudici and R. Castelo, "Improving Markov chain Monte Carlo model search for data mining," *Machine Learning*, vol. 50, no. 1-2, pp. 127–158, 2003.

[17] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, 1969.

[18] D. Endy and R. Brent, "Modelling cellular behavior," *Nature*, vol. 409, pp. 391–395, 2001.

[19] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: *in numero* molecular biology," *Nat. Rev. Genet.*, vol. 2, no. 4, pp. 268–279, 2001.

[20] H. de Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *J. Comput. Biol.*, vol. 9, no. 1, pp. 67–103, 2002.

[21] P. Smolen, D. A. Baxter, and J. H. Byrne, "Modeling transcriptional control in gene networks—methods, recent results, and future directions," *Bull. Math. Biol.*, vol. 62, no. 2, pp. 247–292, 2000.

[22] K.-W. Chu, Y. Deng, and J. Reinitz, "Parallel simulated annealing by mixing of states," *J. Comput. Phys.*, vol. 148, no. 2, pp. 646–662, 1999.

[23] M. Wahde and J. Hertz, "Coarse-grained reverse engineering of genetic regulatory networks," *Biosystems*, vol. 55, no. 1-3, pp. 129–136, 2000.

[24] X. Zhou, X. Wang, and E. R. Dougherty, "Construction of genomic networks using mutual-information clustering and reversible-jump Markov-chain-Monte-Carlo predictor design," *Signal Process.*, vol. 83, no. 4, pp. 745–761, 2003.

[25] X. Zhou, X. Wang, E. R. Dougherty, D. Russ, and E. Suh, "Gene clustering based on clusterwide mutual information," *J. Comput. Biol.*, vol. 11, no. 1, pp. 147–161, 2004.

[26] F. Jensen, *An Introduction to Bayesian Networks*, UCL Press, London, England, 1996.

[27] K. Murphy, "A variational approximation for Bayesian networks with discrete and continuous latent variables," in *Proc. 15th Conference on Uncertainty in Artificial Intelligence*, pp. 457–466, San Mateo, Calif, USA, 1999.

[28] J. H. Albert and S. Chib, "Bayesian analysis of binary and polychotomous response data," *J. Amer. Statist. Assoc.*, vol. 88, no. 422, pp. 669–679, 1993.

[29] K. Imai and D. A. van Dyk, "A Bayesian analysis of the multinomial probit model using marginal data augmentation," *J. Econometrics*, vol. 124, no. 2, pp. 311–334, 2005.

[30] S. F. Arnold, "Gibbs sampling," in *Handbook of Statistics 9: Computational Statistics*, C. R. Rao, Ed., pp. 599–625, North-Holland Publishing, Amsterdam, The Netherlands, 1993.

[31] G. Casella and E. I. George, "Explaining the Gibbs sampler," *Amer. Statist.*, vol. 46, no. 3, pp. 167–174, 1992.

[32] M. Smith and R. Kohn, "Nonparametric regression using Bayesian variable selection," *J. Econometrics*, vol. 75, pp. 317–344, 1996.

[33] K. E. Lee, N. Sha, E. R. Dougherty, M. Vannucci, and B. K. Mallick, "Gene selection: a Bayesian variable selection approach," *Bioinformatics*, vol. 19, no. 1, pp. 90–97, 2003.

[34] C. Robert, "Simulation of truncated normal variables," *Stat. Comput.*, vol. 5, pp. 121–125, 1995.

[35] A. Arkin, P. D. Shen, and J. Ross, "A test case of correlation metric construction of a reaction pathway from measurements," *Science*, vol. 277, pp. 1275–1279, 1997.

[36] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, 1998.

[37] E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach, and R. Shamir, "An algorithm for clustering cDNAs for gene expression analysis," in *Proc. 3rd Annual International Conference on Computational Molecular Biology*, pp. 188–197, Lyon, France, 1999.

[38] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nat. Genet.*, vol. 22, no. 3, pp. 281–285, 1999.

[39] X. Zhou, X. Wang, R. Pal, I. Ivanov, M. Bittner, and E. R. Dougherty, "A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks," *Bioinformatics*, vol. 20, no. 17, pp. 2918–2927, 2004.

[40] C. Andrieu, N. de Freitas, and A. Doucet, "Robust full Bayesian learning for radial basis networks," *Neural Computation*, vol. 13, no. 10, pp. 2359–2407, 2001.

[41] N. Friedman, K. Murphy, and S. Russell, "Learning the structure of dynamic probabilistic networks," in *Proc. 4th Conference on Uncertainty in Artificial Intelligence*, pp. 139–147, Madison, Wis, USA, 1998.

[42] A. A. Alizadeh, M. B. Eisen, R. E. Davis, et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

[43] M. Bittner, P. Meltzer, Y. Chen, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.

[44] M. Brun, E. R. Dougherty, and I. Shmulevich, "Attractors in probabilistic Boolean networks: steady-state probabilities and classification," preprint, 2004.

[45] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[46] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young, "Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks," *Pac. Symp. Biocomput.*, vol. 6, pp. 422–433, 2001.

[47] S. Kim, H. Li, E. R. Dougherty, et al., "Can Markov chain models mimic biological regulation?," *Journal of Biological Systems*, vol. 10, no. 4, pp. 337–357, 2002.

[48] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer-Verlag, New York, NY, USA, 1999.

[49] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[50] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.
[51] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proc. IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

Xiaodong Wang: Department of Electrical Engineering, Columbia University, New York, NY 10027, USA

Email: wangx@ee.columbia.edu

Dimitris Anastassiou: Department of Electrical Engineering, Columbia University, New York, NY 10027, USA

Email: anastas@ee.columbia.edu

Dong Guo: Department of Electrical Engineering, Columbia University, New York, NY 10027, USA

Email: guodong@ee.columbia.edu

10 Compressing genomic and proteomic array images for statistical analyses

Rebecka Jörnsten and Bin Yu

Information technology advancements are bringing about innovations for genomic and proteomic research. One such innovation is the array imaging technology based on which gene or protein expression levels are derived. These images have a fundamentally different purpose to serve than the traditional still images: they are for statistical information extraction, not for visual inspection or comparison. Due to the huge quantity of such images and the limited bandwidth for their sharing among different researchers, for both storage and transmission goals, these images need to be compressed. Dictated by the statistical analyses to follow, in this chapter we lay out a multilayer data structure as the principle for both lossless and lossy compression of array images. We illustrate this principle in the example of cDNA microarray image compression with results of an average of near 2 : 1 lossless compression ratio and an average of 8 : 1 lossy compression ratio. The lossless ratio is comparable with the off-the-shelf lossless compression scheme LOCO, but with the added benefit of a handy structure for statistical analysis; the lossy ratio is obtained with a quantization noise level comparable to that of the imaging technology or the variation between two replicate imaging experiments.

10.1. Introduction

We live in an exciting era of technology innovations with all their advantages (and disadvantages). These innovations are fueling, if not driving, the progresses in genomic research (the study of genetic material such as DNA and RNA), and the newer proteomics research (the study of proteins which are directly responsible for actions in cells).

A revolutionary innovation has been the DNA microarray imaging technology for genomic research and it takes different forms: cDNA (P. Brown, http://www-genome.stanford.edu/), Affymetrics gene chips (http://www.affymetrix.com/index.affx), and Inkjet (http://www.rii.com). It provides measurements of mRNA (messenger RNA) material existing in cells to develop an understanding of gene function, gene regulation, and gene interaction through a simultaneous study of expression levels of thousands of genes. Microarrays are also used extensively in

clinical research. The goal is to identify disease genes, develop treatments, and diagnostic tools. The purpose of a microarray image is not for visual inspection as in still and video image signal processing, but for extraction of statistical information regarding the gene expression levels. The statistical inference problems based on extracted gene expression data are plentiful, and often nontrivial. Among these problems are the identification of important genes, and groups of genes that may be linked in terms of functionality.

We have recently seen the emergence of the new research field called Proteomics, where the focus is shifted from mRNA measurements to proteins [1]. Even though mRNAs carry the genetic instructions to produce proteins, that is, an indirect measurement of protein levels, the proteins themselves are the direct agents to make cells function. Hence, direct measurements of proteins are believed to lead to a better understanding of biologically related events in organisms. To measure the protein amount, antibody materials are put in the spot wells on an array for proteins to bind. This protein array technology is still in its infancy and is more complicated than the DNA microarray technology due to the inherent properties of proteins. One major difficulty lies in the logistics and practical procedures in generating thousands of high-quality probes which have been successfully produced for DNA microarrays. Despite the fact that DNAs (mRNAs) and proteins are very different materials, the protein array images share similar characteristics of DNA microarray images and the same statistical information extraction is the purpose, not visual inspection. In Figure 10.1c we display a small portion of a 10 Mb protein array image, kindly provided to us by Dr. Claudio Caamano, Mental Health Research Institute, University of Michigan.

Both DNA and protein array images contain expression spots laid out in a grid (see Figure 10.1). At the onset of an array experiment the spots contain the known mRNA or antibody materials from DNA or antibody libraries. At the conclusion of the experiment, the corresponding materials from sample cells are added to the spots and allowed to hybridize. Through this technique we can detect the amounts of mRNA or protein in the sample cells. The finished arrays are then scanned to produce array images usually in a 16 bits/per pixel format. All these array images, DNA or protein, are very large (tens of Mb). We see an ever increasing demand on using array technology for genomic or proteomic research in universities, research institutions, and private companies. This increase results in a huge quantity of raw array images that researchers store after a certain processing to extract statistical information. Storage is necessary since the images are expensive to obtain and the processing techniques are still under development. It is safer to keep the raw images for possible reprocessing and improved information extraction as and when new processing techniques become available. Since the quantity of such images is huge, it is well worth the efforts to compress them before storage amid the ever falling hard disk price. To best facilitate the statistical analyses downstream, serious considerations are required to address the question of how to compress, if lossless, and how much and what to take out, if lossy. This chapter proposes principles to answer these questions and illustrates the principles in the example of cDNA microarray image compression.

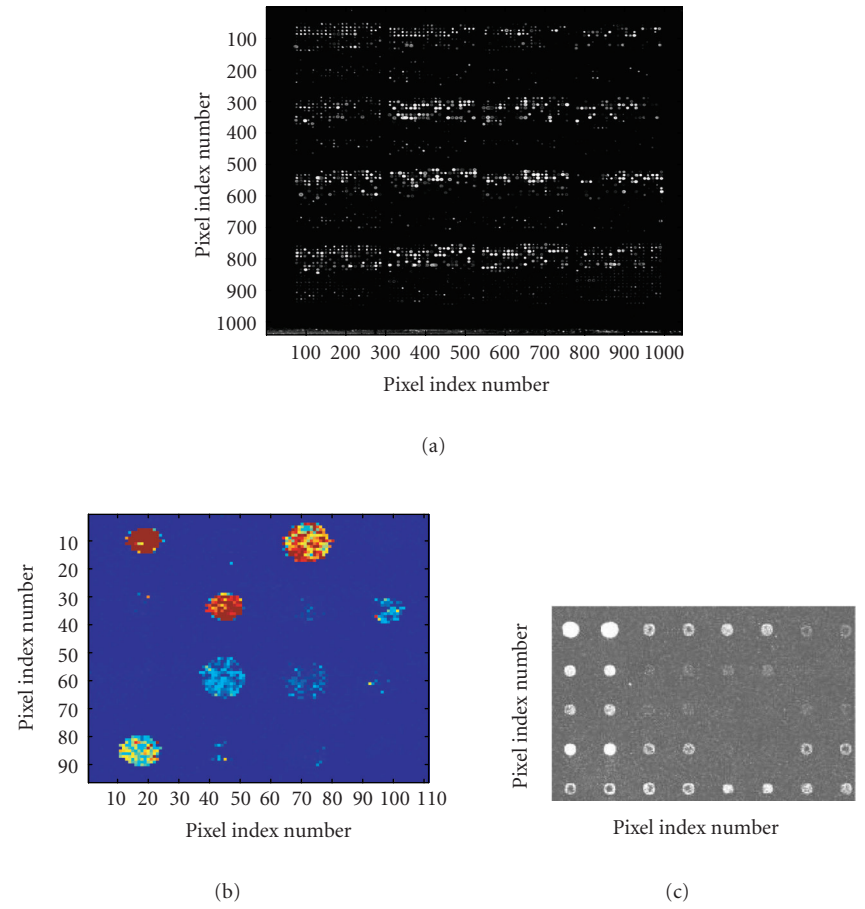


Figure 10.1. (a) Microarray image: 4×4 grid setup, 19×21 spots/grid. (b) A subset of the same image. (c) A subset of a protein array image.

The rest of the chapter is organized as follows. In Section 10.2, we put forward the importance of a multilayer data structure to ease statistical analysis. Lossy and lossless compression is dealt with in Section 10.3 which advocates a multilayer progressive data structure for compressed images. We propose using a variation measure between two replicated images as the desirable quantization noise level in lossy compression, and a partially progressive scheme for lossless image reconstruction. Our multilayer encoded data structure has also recently been adopted by Hua et al. [2] in the context of wavelet-based microarray image compression. In Section 10.4, all our proposals are implemented in the example of cDNA microarray image compression (see [3, 4]). For a set of microarray images kindly provided by M. Callows via T. Speed's group (see [5, 6, 7]), we obtained an average of 2 : 1 compression ratio for lossless compression and an average of 8 : 1

compression ratio for a multilayer progressive lossy scheme with a quantization error comparable to the average difference between two replicated images.

10.2. Lossy and Lossless compression through a multilayer data structure

Images can be completely recovered through lossless compression. Therefore, no one should argue that it is not needed in the face of the facts that the savings on the cost of disks to store raw array images could easily run in millions of dollars. Furthermore, transmission bandwidth is always limited if the images are to be shared by different research groups through internet or a centralized array image library. Even the off-the-shelf lossless compression tools such as LOCO [8] are very useful. Usually it can cut down the cost to half for the current generation of cDNA microarray images. There is not much hope for a lossless compression ratio exceeding 2 : 1 due to the fact that the last 8 bits of each pixel value predominantly contains noise [3].

This treatment of array images as if they are still images does not make the uncompression of them easy for statistical analysis or partial reprocessing later. The whole image has to be decoded for a researcher to get the gene expression levels or later go back to look at even one gene or antibody spot to revise the estimated gene expression level when necessary. As discussed in [3], one would like to have the compressed image in a data structure that is easy for downstream statistical analysis and possible partial reprocessing at a later time.

So, what is a desirable data structure for statistical analysis? First of all, any statistical analysis based on array images needs expression levels of the mRNAs or proteins. These levels are obtainable from the intensity readings of the images (which are related to the amount of dyed mRNA or protein materials) in the mRNA or antibody spots. To get these expression levels, one has to decide where the spots are, that is, a segmentation of the image is needed. Furthermore, the nonspot regions of the images also contain important information. The background part is used for local estimation of intensity drifts which could influence the expression level estimation in a systematic way.

A multilayer data structure for compression

(i) The most needed statistical information should be the first layer in the data structure, that is, the estimated expression levels for each spot (mRNA/protein). For quality control purposes, we also include spot standard deviations in this layer. The standard deviations are measures of spot variability or heterogeneity.
(ii) The second layer of the data structure should contain the segmentation map identifying the spot regions.
(iii) The third layer contains the entropy coded intensities in the spot regions, to a chosen level of precision (lossy compression).
(iv) The fourth layer contains the run length coded intensities in the background regions, to a chosen level precision (lossy compression).

(v) The last layer contains the residual information, for adaptive refinement of spot and background regions to any level of precision (progressive lossy-to-lossless compression).

A statistician in possession of a compressed image in terms of the data structure as described above would have a quick access to the expression levels from the first data layer and some rough variability measures. If she or he is curious about how good the quality of the image is, the segmentation map at the second layer would give some important clues (smooth spot boundaries indicate good quality and ragged ones poor quality). If an expression level is unusual, it calls for the inspection of the corresponding spot and this does not require the decoding of the whole image, only the object corresponding to that spot. For a systematic drift worry, she or he could zoom into the fourth layer directly and uncompress only that part.

10.2.1. Lossless compression

For lossless reconstruction, the researcher can decode all layers. Moreover, the lossless compression ratio for each array is a measure of array data quality. Noisy arrays have highly random least significant bits, which results in a higher lossless bit rate. Therefore, the file sizes of the compressed images can be used to sort the images in terms of data quality.

10.2.2. Lossy compression: what to take out, how, and how much

Since the multilayer coded data structure was dictated by the statistical analysis, a useful lossy compression scheme should follow the same structure. Since the first two layers contain only summary statistics for each gene, the encoding cost of these layers is only marginal compared to the full image data. Since the information contained in these two layers is crucial to the statistical analysis to follow, they should be kept as they are. The next two layers are subject to quantization or lossy compression. Acceptable loss is not readily defined for array images. Recall that these images are not preserved for the purpose of visual comparison, but for statistical information extraction and processing. We define acceptable loss as the level of replicate experiment variation, that is, the level of noise (scanner noise and otherwise) in the images.

In the following section, we illustrate our structured lossless and lossy compression scheme on cDNA microarray images, but emphasize the fact that similar approaches are also appropriate for other types of array experiments.

10.3. An example: cDNA microarray image compression

The cDNA microarray image technology is a tool geared at measuring the "activity" or expression of a gene. A gene is a segment of DNA that maps into a specific protein. The expression of a gene is a two-stage process whereby the protein product is created. The first stage is *transcription* where the DNA segment is translated

into a messenger (m)RNA. The second stage is *translation* where mRNA is used as a blueprint for a protein. Microarray experiments measure the level of activity of a gene at the first stage. Thus, the abundance of mRNA in a cell is related to the amount of the corresponding protein being produced. We measure the abundance of mRNA in a specific sample, *relative* to another sample. DNA probes (each corresponding to a gene, or DNA segment) are placed, or "spotted" onto a microscopic glass slide by a robotic *arrayer*. A reference sample of mRNA is labeled with a green fluorescent dye (Cy3). The sample of interest is labeled with a red dye (Cy5). The two mRNA samples are mixed and allowed to hybridize onto the array. The relative mRNA abundance (for each probe) is measured through the competitive hybridization of the two samples. A laser scan of the array produces two fluorescent intensity images. The intensity ratio for each probe, or spot, is proportional to the relative abundance of mRNA in the two samples. The raw microarray image data thus consist of two high precision (16 bpp) scans. The images are structured, with high intensity spots (corresponding to the probes) located on a grid (see Figure 10.1). The spots are submerged in a noisy and nonstationary background. The spots have roughly circular shape. The background (nonspot regions) can be corrupted by high intensity speckle noise from dust particles, or water droplets on the glass slide. Spots may "bleed" into each other or be smeared due to imprecision in the spotting procedure, or through "washing-out artifact," as excess sample material is removed from the array prior to scanning.

10.3.1. Genetic information extraction

Since relative mRNA abundance is measured in microarray experiments, the genetic information quantity available is the *differential gene expression* between the two samples. In order to accurately estimate this quantity, we have to identify the high intensity regions in the images corresponding to each probe, and where hybridization has occurred. Moreover, we have to estimate, and correct for, the local background intensity or noise level. Various methods for image segmentation and background correction are used in the processing of microarray images. These methods have had variable success, depending on noise level, average spot sizes, and distances, of the arrays (e.g., [7] and M. S. Eisen, http://rana.stanford.edu/software).

Segmentation. Automatic registration of the image is used to determine the approximate centers, or the grid location, of the spots. The spots are somewhat circular. The most simplistic approach to identify the regions where hybridization has occurred is through a fixed circle segmentation (http://rana.stanford.edu/software). A circle, with radius chosen from the estimated grid structure, is used to define the spot regions. The apex of the circle is adjusted locally to maximize the summed signal intensity within the circle. When spot sizes vary across the array (see Figure 10.1), which can result from, for example, variations in print-tip pressure, an adaptive circle segmentation is more appropriate [9]. This allows for differences in spot radii, and can significantly improve the identification of spots.

When spot shapes are highly noncircular, adaptive shape segmentation techniques, such as seeded region growing, can improve spot regions identification further still (e.g., [7]). Here, we use a seeded region growing algorithm for initial segmentation, followed by a two-component Gaussian mixture model fit, to further refine the boundaries of the spots, or regions of interest (ROI). Seeds for background and ROIs are available from the registration procedure.

Background correction. Obtaining estimates of the local background intensity level can be a difficult task. When the arrays are noisy, and the spots are positioned close together, the background estimates will often be highly variable. Background estimates are commonly obtained through sampling of the background regions, for example, by looking at disk shaped regions near the identified ROIs (http://rana.stanford.edu/software), or in the "valley-between-peaks" (regions that are the most distant from the center of gravity of all surrounding ROIs) [9]. These methods work well if the spots are clearly separated, but may otherwise perform poorly. A more robust background estimation scheme is presented in [7], and relies on filtering operations (erosion and dilation). This method exhibits low variance, but tends to underestimate the local background intensities.

Summary statistics. The summary statistic of main interest is the estimated differential gene expression, commonly measured on a $\log_2$-scale. Pixels in each image are summed within each ROI. We denote by R_i the red (fluor tag) scan pixels, and by G_i the green scan pixels. The differential expression level, $\log_2(\mu_R/\mu_G)$, is then calculated as the log-ratio of the mean ROI intensities:

$$\log_2\left(\frac{\mu_R}{\mu_G}\right) = \log_2\left(\frac{(1/S)\sum_{R_i\in\mathrm{ROI}} R_i - \sigma_R}{(1/S)\sum_{G_i\in\mathrm{ROI}} G_i - \sigma_G}\right), \tag{10.1}$$

where σ refers to the estimates of the local background and S is the number of ROI pixels. The log-ratio, commonly referred to as M, $M = \log_2(\mu_R/\mu_G)$ is used in downstream statistical analyses, such as clustering and classification.

Other summary statistics of interest are measures of quality, such as spot variances, spot shapes, and product intensities $A = \log_2\sqrt{\mu_R\mu_G}$. The product intensities, A, are often indicative of how reliable the measurements of the differential gene expressions are.

Normalization. Normalization is necessary prior to downstream analysis, since otherwise systematic variation in the data may dominate over chance variation. The fluorescent dyes used to label the two samples have different labeling efficiencies, and there is a dye bias in scanning efficiency. Furthermore, there are spatial systematic errors, for example, a print-tip in one part of the array may show a significantly higher red intensity than in other parts of the array.

It is common to use the genes that show little differential variation between samples for normalization. In some experiments, most genes are not differentially

expressed, and all spots can be used for normalization. In other experiments, a set of housekeeping genes are used. M and A are simply a coordinate transformation of the background corrected spot intensities μ_R and μ_G. Dye-bias normalization is usually conducted on M, with respect to A.

Normalization is still an active area of research. The challenge remains to remove systematic bias while keeping increased variance at bay. In addition, normalization procedures can in some cases introduce bias for low-expression spots. Here, we apply the normalization scheme of Yang et al. [5]. The normalization is nonlinear, and print-tip specific. For each print-tip on the array (one of the 4×4 subgrids of Figure 10.1), we estimate a locally linear fit of M on A. The residual vector $\widetilde{M}$ is computed, and used as the spatial and dye-bias corrected differential gene expressions in subsequent analysis.

In the subsequent statistical analysis, for example, identification of important genes, both $\widetilde{M}$ and A may play a role. An overview of some methods for gene identification based on $\widetilde{M}$ only, or both $\widetilde{M}$ and A, can be found in [5] (see also [10, 11, 12]). Classification and clustering of gene expressions and samples uses the information in $\widetilde{M}$ only. Since image processing, background correction, normalization, and analysis methods are still under development, an image compression scheme for microarray images evidently has to preserve information on both M and A.

10.3.2. Lossy and lossless compression of cDNA microarray images

Lossless compression of microarray images is easier for experimentalists to accept. However, for efficient transmission of image data for data sharing, we need to consider the use of lossy reconstructions of the images for subsequent analysis. Not all processing steps require lossless image reconstructions. In addition, the equivalence of compression and denoising suggests that we can obtain improved genetic information extraction from the images with lossy reconstructions (Section 10.4). Segmentation is a relatively easy task and can be done on crude 8 bpp reconstructions (see [3]). Background correction and normalization are more sensitive to information loss, especially in the region of low intensity spots. A successful compression scheme thus has to keep more precision for low intensity spot regions, but can use a coarse image reconstruction for high intensity spot regions. Note that the commonly used criterion for lossy compression, mean squared error, does not reflect this requirement.

The variance introduced by compression is much more of a concern than marginal bias in multistep processing (segmentation, background correction, and normalization). Our aim is to keep both bias and variance under control, and ensure that the effect of compression is smaller than the variability between replicated experiments. We define this as *acceptable loss* for microarray image compression. For our sample image dataset this acceptable loss corresponds to a bit rate of ~4 bpp (compared to 32 bpp with no compression) (Section 10.4).

Lossless and lossy compression of natural and medical images is a mature field in the engineering sciences. However, the performance of state-of-the-art

image compression schemes on microarray data is poor. Here we mention some of the issues that make the compression of microarray images a particularly difficult problem. Microarray images are very noisy, more so than images in many medical applications. We can compare the lossless compression rate of microarray images to those reported for mammograms. Lossless compression ratios of mammograms are ∼3:1, with the very efficient SPIHT algorithm (see [13, 14]), whereas only ∼1.6:1 for microarray images. In medical imaging, the final task is visual inspection. Lossy compression of medical images thus often takes the approach of defining an ROI, within which high precision lossy reconstructions are used. Outside the ROI, a coarse image reconstruction is used, and this leads to considerable savings in overall bit rate. For microarray images, the non-ROI regions cannot easily be discarded as less informative, or be preserved at an arbitrary low bit rate. These regions are used for background intensity estimation. In medical imaging, the ROIs are usually large, few (1–2), and arbitrarily picked by the experimentalists (a square or circle centered at an ROI). ROIs in microarray images are the spot regions. These are defined by the segmentation, and have distribution characteristics that differ from the background. There are many ROIs (thousands) for each microarray image. They are located close together, and can be of arbitrary shape. They are small, with an approximate diameter of 8–16 pixels. The small ROIs precludes the use of off-the-shelf wavelet-based compression schemes, which tend to use wavelets with relatively wide support intersecting with multiple spots. The many and small high intensity regions create large wavelet coefficients over almost all the image subbands. At low bit rates, algorithms such as SPIHT, or wavelet and zero-tree coding [13], will be dominated by the edges around the high intensity spots. In [2], an adaptive wavelet transform is used to alleviate this problem. Compression schemes that are not wavelet based, but based on predictions in the spatial domain, also have difficulty with the many high intensity spots. A rowscan-based prediction scheme creates a "smearing" bias in the image reconstruction. Thus, it is a nontrivial task to employ the principles of lossless ROI coding, or coding of ROI and non-ROI at different precision, used in medical imaging, to microarray images (see [14, 15]).

We here choose to take a spatial prediction approach. To avoid the "smearing" bias we encode the spot regions and the background separately. We first transmit an *overhead* defining the ROI and background, that is, a segmentation map. We refer to this approach as segmented LOCO (SLOCO) (see [4]).

Segmented LOCO—SLOCO. Our scheme builds on the JPEG-LS lossless standard, LOCO (LOw COmplexity), see [8]. A low complexity scheme is preferable for this application. The characteristics (size, shape of spots, level of noise, background drifts, and artifacts) of microarray images can be very different depending on which lab produced the data. It is therefore near impossible to come up with an advanced compression scheme that works on images from different labs. To ensure that the compression scheme performs reasonably well for images from many labs, simplicity is key. Below, we outline the components of LOCO and SLOCO (details can be found in [3, 8]).

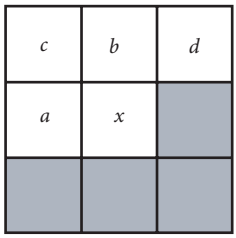


Figure 10.2. Causal context of pixel x (a, b, c, d).

Fixed and adaptive prediction, context modeling. The LOCO algorithm uses a robust pixel domain prediction scheme, for example, the causal context (Figure 10.2) of pixel x is used to predict its value. The initial fixed predictor is

$$\hat{x}_{\text{fix}} = \text{median}(a, b, a + b - c). \tag{10.2}$$

It works on the same principle as a simple edge detector. Note that a, b, and c are *lossy* reconstructions of the corresponding pixel values. We have briefly studied the use of more complex predictors for microarray images, but have seen no gain with the use of a more complex predictor. The different noise level and array characteristics of arrays from various labs make the building of robust predictors with more structure (e.g., gradient, surface fitting component) extremely difficult.

To further improve on the fixed prediction scheme, LOCO also uses a context-based *adaptive predictor*. The contexts are defined by the vector of local gradients. The local gradients, $g1 = d - b$, $g2 = b - c$, $g3 = c - a$, are quantized to, for example, 16 levels. Each triplet of quantized gradients forms a context class. Based on the past performance of the fixed predictor, within each context class, an estimate of the prediction bias, $\hat{R}$, is obtained. The bias is taken as integer valued and is estimated as follows. The bias estimate of the context is initially set to 0. We assume the current context has been called N times during the coding of the image. Each time we encounter this context the accumulated prediction error B (after adaptive prediction) is compared to N. If $B < -N$, we set $\hat{R} = \hat{R} - 1$. If $B > N$, we set $\hat{R} = \hat{R} + 1$. We do not allow an absolute $\hat{R}$ value greater than a preset $R_{\max}$. The adaptive predictor is

$$\hat{x} = \hat{x}_{\text{fix}} + \hat{R}. \tag{10.3}$$

Quantization and encoding of prediction errors. The standard LOCO algorithm has been extended to near-lossless compression [8]. There, the prediction errors are quantized with a uniform quantizer (UQ) with bin widths $2\delta + 1$, and reconstruction at the center of each bin. If δ is small, and the number of quantization bins is large, the uniform quantizer is close to the MSE distortion optimal quantizer, for an extensive family of distributions (see [8, 16]). The uniform quantizer also puts a bound on the maximum pixelwise error, that is, δ. If δ is large, the UQ is far from

optimal from MSE distortion point of view. The optimal quantizer, subject to an entropy constraint, is indeed uniform for a large family of error distributions, but the reconstruction levels are not at the center of the quantization bins. In SLOCO, we use a UQ-adjust quantizer for large δ. Thus, the bins near the center of the error distribution, where most of the probability mass is located, have adjusted reconstruction levels closer to the MSE distortion optimal, as well as smaller bin widths $2\delta' + 1$, such that the maximum error for all pixels is still bounded by δ. The outer bins have bin widths $2\delta + 1$, and center bin reconstruction levels, that is, given a prediction error $\epsilon = x - \hat{x}$, the error is quantized using the UQ-adjust quantizer. If the K center-most bins have bin width $2\delta' + 1$ the quantized error is

$$Q(\epsilon \mid |\epsilon| \leq (2K+1)\delta' + K)) = \text{sign}(\epsilon)\left\lfloor \frac{|\epsilon| + \delta'}{2\delta' + 1} \right\rfloor \tag{10.4}$$

and

$$Q(\epsilon \mid |\epsilon| > (2K+1)\delta' + K)) = \text{sign}(\epsilon)\left(K + \left\lfloor \frac{|\epsilon'| - 1}{2\delta + 1} \right\rfloor\right), \tag{10.5}$$

where $\epsilon' = \text{sign}(\epsilon)(\epsilon - (2K+1)\delta' + K)$.

As in LOCO, we use Rice mapping to map the quantizer error distribution into a smaller alphabet without loss of information. The distribution of the mapped and reduced range quantized prediction errors is quite close to a one-sided geometric distribution, and encoded with a Golomb code [8]. A quantizer bin index y is thus encoded in two parts. Given a code parameter $m = 2^k$, we first encode in unary the most significant bits of y, $\lfloor y/m \rfloor$. The remainder $y \bmod m$ is encoded in binary representation. The total code length for y is thus $k + 1 + \lfloor y/2^k \rfloor$ bits. For optimum performance, we use many Golomb codes for the encoding of the errors. Each context class builds its own code. The Golomb parameter k that best matches the context distribution is estimated adaptively, by $k = \arg\min_{k'}\{2^{k'} \geq \bar{A}\}$. Here, $\bar{A}$ is the mean absolute quantized (and reduced range) prediction error of the context class.

We need to update the context parameters for optimal Golomb coding. We have already mentioned how the integer bias is updated each time a context is encountered. Other context variables we store are the accumulated prediction errors after range reduction (but before Rice mapping). We also store the accumulated absolute prediction errors, as well as a counter for each context. After a context has been encountered T times, we reset all context variables to half their current value. This keeps the context information current. We use $T = 64$.

Run length coding. LOCO encodes the smooth regions of an image efficiently by means of a run length code. If the quantized local gradients indicate a smooth region ($all(g1, g2, g3) \leq \delta$), we predict the value a (i.e., the causal horizontal neighbor of x) for x. Furthermore, we predict that $l - 1$ pixels following x also equal a. Instead of encoding each pixel separately, we only have to encode the deviation of the length of the "run" of values $a \pm \delta$, from the expected run length l. The length of the observed run is encoded with a Golomb code with parameter g,

where $l = 2^g$. When we encounter a value different from $a \pm \delta$, we encode the interruption pixel using similar techniques as for the encoding of regular pixel samples. We use two contexts for the interruption pixels, context 1 is used if $a = b$, and context 2 if $a \neq b$. The expected run length l is updated based on the recently observed run lengths. For each run equal to l, a counter index is incremented. Once the counter index exceeds a threshold, the expected run length is increased by one unit. Similarly, the expected run length is decreased after a number of interruption have been encountered.

Overhead. The overhead of the SLOCO algorithm contains the spot means and standard deviations, as well as the local background intensity estimates and the local background standard deviation. The overhead also contains the estimated grid structure of the microarray images, and the segmentation map.

The spot and background means are encoded using adaptive Lempel-Ziv (LZ). The cost of the spot mean overhead amounts to approximately 11–15 bits/spot on the images we have examined. The background means have a much smaller range and are easily encoded using only 3–5 bits/spot. The spot variance is approximately proportional to the spot mean. Conditioning on the spot mean, we can encode spot and background standard deviations with 5–7 and 4–5 bits/spot, respectively. The segmentation map is efficiently encoded using the chain code of Lu and Dunham [17]. Using a seeded region growing algorithm, the shapes of the spots can be quite arbitrary, and cost between 1.2–1.6 bits times the circumference of the spot to encode. The average cost of the overhead for the images we examined is 0.376 bpp.

If no reprocessing of the images is needed, the overhead contains all relevant information for downstream analysis. In addition, it contains spot quality measurements such as spot shapes, variances, and the local background variance.

Coding the spot regions. Given the overhead, we can compute the signal-to-noise ratio of each spot. We use the following signal-to-noise ratio measure:

$$\mathrm{SNR} = \frac{m_{\mathrm{spot}}^2}{m_{\mathrm{bg}}^2 + s_{\mathrm{bg}}^2}, \tag{10.6}$$

where $m_{\mathrm{spot}}, m_{\mathrm{bg}}$ denote the mean spot and background intensities, and s_{bg}^2 the estimated background variance.

Based on the SNR, we can pick a bound on the maximum pixelwise error δ for each spot (see below). The size of each spot is too small to allow for any adaptive prediction step, or for adaptive estimation of the Golomb parameter. We therefore use a fixed Golomb code, and only the fixed predictor $\hat{x}_{\mathrm{fix}}$ within each spot. The spot Golomb parameter could be estimated on the encoder side, after applying the fixed predictor, and transmitted as overhead to the decoder. However, we can do nearly as well by using an approximate estimate of the optimal Golomb parameter k obtained as follows.

The overhead contains the spot standard deviation. The expected value of the absolute prediction error is well approximated by $s_{\mathrm{spot}}/(\sqrt{2} \times 1.3)$, where s_{spot} is

the estimated spot standard deviation. The factor 1.3 has been estimated from the microarray image data. The fixed predictor is local and gives smaller prediction errors than using the spot mean as a global predictor. Given the bound on maximum pixel error δ, we thus estimate k as

$$\hat{k} = \max\left(0, \left\lceil \log_2\left(\left\lfloor \frac{\bar{A}/1.3 + \delta}{2\delta + 1} \right\rfloor\right)\right\rceil\right), \tag{10.7}$$

where $\bar{A}$ is the MAD estimate of s_{spot}, divided by $\sqrt{2}$. We encode the spots in a row scan manner. When the fixed predictor is applied, missing context pixels are imputed with the spot mean value from the overhead.

Coding the background. The background is encoded in a row scan fashion for subblocks of the images. Larger blocks are more efficient to encode, but we find that encoding 4×4 blocks for the background for each print-tip configuration gives approximately the same bit rate as encoding the background as one solid block. This allows image subset reconstruction.

To use the fixed predictor, we impute the missing context pixels (the spot pixels) with a value equal to 3 times the local background intensity estimates (from the overhead). The factor 3 provides a gradient near the spots. Imputing the missing context pixels with the background intensities gives a lower bit rate, but including a factor of 3 reduces the prediction variance.

Given the spot SNR values near a specific background region and the local estimate of the background intensity, we pick a bound on the maximum error δ (see below). The background region of a spot consists of pixels closer to this spot than any other. The context Golomb parameter k is estimated as in standard LOCO. Because in SLOCO the maximum error bound varies from region to region, we need to store the reconstructed prediction errors, not the quantization bin indices as in LOCO. The context parameter is estimated as $\hat{k} = \max(0, \lceil \log_2(Au(Q)/N(Q)) \rceil)$, where $Au(Q) = \lfloor A(Q)/(2\delta'' + 1) \rfloor$, $N(Q)$ is the context counter and $\delta'' = \delta$ for the K centermost quantization bins, and δ otherwise. $A(Q)$ is the context variable containing the accumulated absolute prediction errors, after reconstruction. If the context of pixel x indicates a smooth region, we apply the following run length coding strategy.

The run length coding of SLOCO differs from that of LOCO. We do not allow runs to cross from a region with higher maximum error bound δ into one with smaller δ. If a spot is encountered during a run, we skip ahead to the next background pixel. However, we do not necessarily continue the run. We denote the current pixel by x. If x is on the boundary of a spot, we skip ahead to the next background pixel in the same row as x and denote this pixel by y. If $\delta(x) \leq \delta(y)$, we compute the vector of local gradients of x and y; $\tilde{g}(x)$ and $\tilde{g}(y)$. If $\max |\tilde{g}(x) - \tilde{g}(y)| \leq \delta(y)$, we continue the run. Note that in this case the gradients are computed with imputed context values equal to the local background estimates, *not* with a factor 3. Runs interrupted by a decrease in δ, or by $\max |\tilde{g}(x) - \tilde{g}(y)| > \delta(y)$, are encoded as "expected interruptions" by appending a 1 to the bitstream. Other interruptions are encoded in the same manner as in standard LOCO. The criteria

for continuing or interrupting a run are available at the decoder from causal information, or from the overhead.

Maximum error bounds. We choose different δ for the background regions because we recognize the need for higher precision near low intensity spots. We use thresholds on the SNR values to pick the maximum error bounds. For the spot regions, we use $\delta = \{511, 255, 127, 63\}$. The $\log_2$ SNR thresholds corresponding to each δ level are chosen as the quantiles $(90, 50, 10\%)$ of the SNR distribution of the array images.

The near lossless LOCO design scheme assumes that δ is small and that the number of quantization bins is large, such that the quantization errors are approximately uniformly distributed within each quantization bin. For the 16 bpp images, using a large $\delta = 255$ still corresponds to a considerable number of quantization bins. However, the error distribution is very "peaked" in the innermost quantization bins. Depending on the δ used, as discussed above we let the $K(\delta)$ innermost quantization bins have width $2\delta' + 1$, where $\delta' < \delta$. The corresponding reconstruction levels are adjusted such that the maximum error is still bounded by δ. We pick K and δ' such that the increase in the number of quantization bins is small compared to the UQ quantizer used by standard LOCO. For $\delta = \{511, 255, 127\}$, we use $\delta' = \{444, 224, 115\}$ and $K(\delta) = \{6, 10, 20\}$. For smaller δ, the UQ is used. With this setup, the additional number of quantizer bins for the 16 bpp images is between 3 and 5.

Flowchart of SLOCO

Coding the spot regions

(0) Initialize: given the spot SNR, set the maximum error bound to equal $\delta(\text{SNR})$. Estimate the Golomb parameter k by

$$\hat{k} = \max\left(0, \left\lceil \log_2\left(\left\lfloor \frac{\bar{A}/1.3 + \delta}{2\delta + 1} \right\rfloor\right)\right\rceil\right), \tag{10.8}$$

where $\bar{A}$ is the mean absolute value of (spot pixels − spot mean) $\simeq s_{\text{spot}}/\sqrt{2}$.

(1) For current pixel x, apply the fixed predictor, replacing missing pixel values with the spot mean. Compute the prediction error ϵ.

(2) Quantize ϵ using the UQ-adjust quantizer. Reduce the range of the quantized residuals and apply the Rice mapping $\rightarrow \hat{\epsilon}$.

(3) Encode $\hat{\epsilon}$ with Golomb$(\hat{k})$.

Coding the background

(0) Initialize: set context variables $R, B, N = 0$ and $A = 4$. Using the overhead, pick $\delta(x)$ and $\delta'(x)$ for all background pixels x. R is the integer bias, B the accumulated prediction error, A the accumulated absolute prediction error, and N the context counter. Pick the UQ-adjust parameter K.

(1) For current pixel x, impute nonbackground pixels with 3 times local spot background estimate. Compute gradients $g1, g2, g3$ and quantize each to 16 levels. Denote the context index by Q.
(2) if $all(g1, g2, g3) \leq \delta(x)$ go to run length coding mode, otherwise goes to 3.
(3) Apply the fixed and adaptive predictor to x. Compute the prediction residual $\epsilon = x - \hat{x}_{\text{fix}} - \hat{R}(Q)$.
(4) Quantize ϵ using the UQ-adjust quantizer. Reduce the range of the quantized residuals and apply the Rice mapping $\rightarrow \hat{\epsilon}$.
(5) Estimate the context Golomb parameter k by

$$\hat{k} = \max(0, \lceil \log_2(Au(Q)/N(Q)) \rceil), \tag{10.9}$$

where $Au(Q) = \lfloor A(Q)/(2\delta'' + 1)$, $\delta'' = \delta$ if $|\hat{\epsilon}| > K$, and δ' otherwise.
(6) Encode $\hat{\epsilon}$ with Golomb$(\hat{k})$.
(7) Update all context variables. $B(Q) < -B(Q) + \rho(\hat{\epsilon})$, $A(Q) < -A(Q) + |\rho(\hat{\epsilon})|$, where ρ is the reconstruction levels of the UQ-adjust quantizer.

Run length coding

(0) Initialize: set run count $c_t = 0$. Set the run error variable $\delta_r = \delta(x)$. Set the run value $r = a$, where a is the left neighbor of pixel x.
(1) While $|x - r| \leq \delta_r$ and the class index of x is "background," add 1 to the run count c_t and go to the next pixel.
(2) If $c_t = l$, where l is the expected run length, add a "1" to the output bitstream and reset c_t to 0.
(3) If the class index of x is "ROI," skip ahead to the next background pixel y. If $\delta(y) \geq \delta_r$ and the gradients of y are within $\delta(y)$ of the gradients of x, continue the run. Otherwise, interrupt the run by appending a "1" to the bitstream.
(4) If "end-of-line" is encountered, interrupt the run and append a "1" to the bitstream.
(5) If $|x - r| > \delta_r$, interrupt the run, append a "0" to the bitstream, and encode the length of the interrupted run $(< l)$.
(6) If the run was interrupted in 5, encode the run-interrupt sample x with the two special contexts depending on whether $a = b$ or $a \neq b$, where b is the top neighbor of pixel x.
(7) Update the expected run length parameter l based on the number of runs encoded. Go back to standard coding of the next background pixel.

To summarize the section, SLOCO differs from standard LOCO mainly in these four aspects: (i) the spots and backgrounds are encoded separately to avoid bias by smearing, (ii) a UQ-adjust quantizer is used instead of the UQ quantizer, (iii) we allow for different maximum error bounds δ in the background regions, and (iv) we use a run length code that takes the segmentation map and the varying δ into account.

The varying δ presents us with some difficulties. Ideally, we should use different contexts for each δ. However, we cannot hope to gain enough information to "train" the context-based predictors if we use separate contexts. We limit ourselves to use one set of context variables. Therefore, we need to store the quantizer reconstructions, rather than the bin indices as is done in standard LOCO.

Lossless compression and progressive transmission. As we will see in the next section, lossy reconstruction at bit rate ∼4 bpp may be sufficient for genetic information extraction with a variety of existing methods for information extraction. However, methods are still under development, and there is no guarantee that using 4 bpp image reconstructions will suffice when applying new methods. Therefore, our compression scheme can be extended to a fully lossless reconstruction of the microarray images. Given the initial lossy reconstruction, the image reconstruction can be refined, spot by spot, background region by background region, or even pixel by pixel, to any bit rate above the minimum decodable bit rate. Our scheme is thus partially progressive.

The overhead provides us with the maximum error δ in each region of the lossy reconstruction of the images. We use a simple and flexible way of encoding the residual image. After prediction and quantization, the quantization errors are close to uniformly distributed in the range $[-\delta, \delta]$. We cannot reduce the first order entropy ($\sim \log_2(2\delta + 1)$) much via predictive coding. We therefore encode bit planes of the residual image. We choose this simple coding scheme since it gives us total freedom to encode any part of the residual image at any rate we desire, independently of what we choose to do in other regions of the image. Despite this apparently inefficient code, we get comparable lossless compression results using the SLOCO and bit plane coding, as we get with standard LOCO. However, we stress that standard LOCO is *not* progressive so as to obtain that bit rate the whole image has to be encoded and decoded in full. We considerably improve over the SPIHT algorithm lossless bit rate with our scheme.

10.4. Results and comparison of methods

10.4.1. Image datasets

Our image compression scheme has been tried out on microarray images from three sources. Here, we will discuss the results obtained from a replicate experiment, courtesy of Matthew Callows, Genome Sciences, Lawrence Berkeley National Lab. The replicate experiment allows us to more convincingly evaluate the effects of compression on the statistical analysis of extracted genetic information. A discussion on the statistical analysis of this dataset can be found in [6]. Results on images from P. Brown's lab, Department of Biochemistry, Stanford University, and the Lawrence Livermore National Lab are not shown here in order to conserve space. However, bit rate results were similar to those obtained on the image dataset from the replicate experiment.

The replicate dataset consists of 16 pairs of images; 8 pairs from 8 "control" mice and 8 pairs from a "treatment" group. The experiment is referred to as Apo

Table 10.1. Lossless bit rates.

Method	Bit rate (cmp 32 bpp)	Compression ratio
LZ (gzip)	21.62	1.48 : 1
SPIHT	19.44	1.65 : 1
WT + ZT + EC	18.64	1.72 : 1
LOCO	17.28	1.85 : 1
SLOCO	17.45	1.83 : 1

AI, and is described in greater detail in [5]. Apolipoprotein AI is related to lipid metabolism in mice. The treatment group has a knocked-out AI gene. The reference sample used for both the control and the treatment group is a pooled sample from all 8 control mice. Each image pair provides relative measurements of the expression levels of 6384 probes, 200 of which are expected to be related to lipid metabolism. With replicate experiments, we can evaluate the compression using a number of test statistics. We compare the estimated gene expression levels (after normalization) $\widetilde{M}$ using the full image data, and lossy reconstructions of the images. We also look at the extracted product intensities A.

To conserve space, we present results using two different methods. We compare seeded region growing and Gaussian mixture segmentation to adaptive circle segmentation. We also compare the robust filtering background correction scheme [7] to the valley-between-peaks method [9]. We find that the choice of background correction scheme has the largest impact on performance. Similar conclusions were drawn in [7]. Here, we therefore compare only two combinations; seeded region growing + robust filtering (hereafter referred to as Method 1) and adaptive circle segmentation + valley-between-peaks (Method 2). We can think of the methods as two extremes. Method 1 has low variance but is possibly biased, whereas Method 2 is variable and susceptible to noise in the data.

10.4.2. Bit rate results

For lossless compression we compare the results using our SLOCO + bit plane coding scheme to standard LOCO, and to the SPIHT algorithm. As a baseline for lossy + bit plane coding, we look at a wavelet lossy scheme with zero-tree coding + entropy coding of the residual image (WT + ZT + EC). This method is widely recognized as one of the best in both lossy and lossless compression of natural and medical images. The lossy rate of (WT + ZT + EC) used is the same as for SLOCO. The reported lossless bit rate for this last scheme is optimistic, that is, based on an entropy calculation of the (WT + ZT) residual image. In Table 10.1, bit rates and compression ratios are shown. The results using LZ and SPIHT are dismal. The compression ratios are 1.48 : 1 and 1.65 : 1, respectively. Note that in medical imaging applications, compression ratios with LZ is commonly ∼2:1 and 3 : 1 with SPIHT. The (WT + ZT + EC) has lossless compression ratio 1.72 : 1. The LOCO lossless compression ratio is 1.85 : 1. We get very close to this result with our scheme, 1.83 : 1. This bit rate result includes the overhead (∼0.376 bpp).

We can achieve a better compression ratio than LOCO (1.87 : 1) if we impute missing context pixels with the background estimates, instead of 3 times the values used here. However, this introduces too much variance in the background region of the initial lossy reconstruction. Since we want the lossy reconstructions to act as good substitutes for the full data for the purpose of reprocessing, we sacrifice a lower lossless bit rate for good progressive performance. We cannot hope to achieve much better lossless compression ratios than this with any method. The 8 least significant bits are close to random for microarray images, that is, have marginal entropy 8 bpp, and are unpredictable. This puts a ceiling of 2 : 1 on the lossless compression ratio.

10.4.3. Lossy compression

We compare SPIHT, LOCO, and SLOCO in terms of how well the genetic information is preserved in the *lossy* image reconstructions. We compare the extracted information from *lossy* microarray images at a certain bit rate to the extracted information from the full (lossless) image data. The results obtained with the (WT + ZT + EC) compression scheme are very poor, and this method is dropped from the comparison. The wavelet transform has a wide support, such that the spots are blended with each other.

Figure 10.3a shows the spot product intensity A (on a log scale) extracted from lossy reconstructed images plotted against the product instensitites based on the full data. The solid red line corresponds to the full data (lossless), the blue markers show the SPIHT reconstruction at 4.1 bpp, and the green markers LOCO reconstruction at 4.4 bpp. The LOCO algorithm requires a fixed δ input, and 4.4 bpp was the closest available bit rate to 4.1. Note that the variance of A is high using the SPIHT algorithm, indicated by the wide spread of the blue markers around the red solid line corresponding to the full data. LOCO shows significant positive bias in A for low A values, indicated by the green markers off-center and above the red solid line. This bias is a result of the smearing of coding spots and backgrounds in one segment. In Figure 10.3b the same results are shown with SLOCO (black markers) at bit rate 4.1 bpp results superimposed. Note that SLOCO controls both the bias and variance better than LOCO and SPIHT, respectively, with the black markers centered tightly around the red solid line corresponding to the full data.

Figure 10.4 shows the normalized differential gene expression levels $\widetilde{M}$ obtained from lossy image reconstructions using SPIHT (blue markers), LOCO (green markers) and SLOCO (black markers) at the above bit rates, compared with differential gene expression levels obtained from the full data (red solid line). Note that all schemes generate accurate results for large absolute differential gene expression levels and are variable for $\widetilde{M}$ near 0. We again see that the wavelet-based SPIHT algorithm produce more variable results than LOCO and SLOCO. The SLOCO scheme fairs slightly better than the standard LOCO.

Our scheme is partially progressive. We can opt to add more precision to the regions of the images corresponding to small $\widetilde{M}$, where the effect of compression

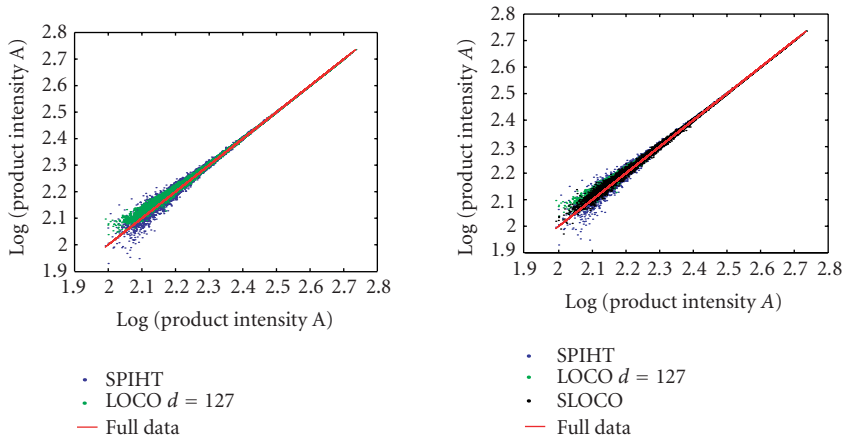


Figure 10.3. (a) Extracted product intensities A from lossy image reconstructions using SPIHT (blue) at rate 4.1 bpp and LOCO (green) at rate 4.4 bpp compared with lossless image reconstructions (red). The blue markers indicate SPIHT is highly variable, whears the green markers show that LOCO is biased for low intensities. (b) Variance and Bias are better controlled with SLOCO at 4.1 bpp as indicated by the superimposed black markers.

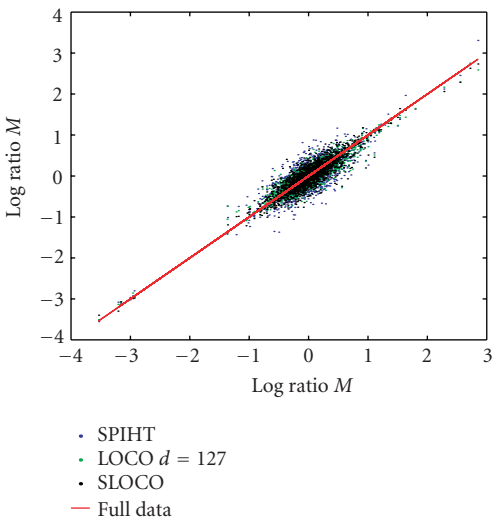


Figure 10.4. Differential gene expression $\widetilde{M}$ based on lossy reconstructions using SPIHT and SLOCO at 4.1 bpp (blue, black) and LOCO (green) at 4.4 bpp compared to lossless reconstructions (red solid line). The lossy methods deviate more from the lossless reconstructions for $\widetilde{M}$ near 0, and generate accurate results for large absolute $\widetilde{M}$.

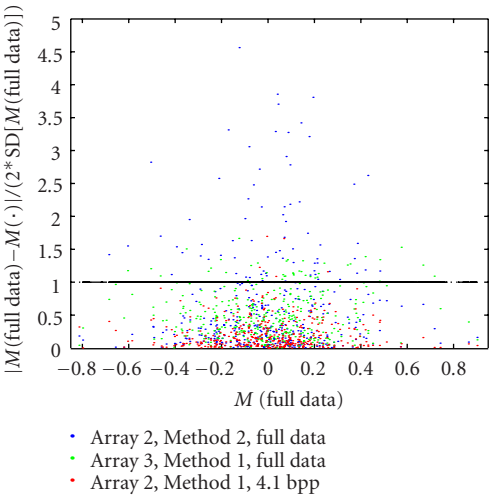


Figure 10.5. "Standardized plot," that is, z-statistic/2. The markers display differences in M from Array 2 using (i) Method 2 on full data (method-to-method variability—blue markers), (ii) Method 1 on Array 3 (array-to-array variability—green markers), and (iii) Method 1 on 4.1 bpp reconstruction of Array 2 (lossy-lossless variability—red markers). Note that the method-to-method variability dominates. The lossy-lossless variability is below the array-to-array variability.

is more noticeable. However, we find that the variability introduced by the lossy compression, in the extraction of the differential gene expression levels, is smaller than the array-to-array variability. We can thus hypothesize that the information "lost" due to compression is below the level of the noise in the data. We also find that the difference in extracted gene expression levels between Method 1 and Method 2 is much greater than the difference between lossless and lossy reconstructions of the images. In Figure 10.5, the baseline for comparison is the gene expression levels extracted from Array 2 using Method 1. We compare the results we get using Method 2 on the same array (blue markers—method-to-method variability), and Method 1 on Array 3 (green markers—array-to-array variability) with the full data. We also compare the results we get using Method 1 on a 4.1 bpp lossy reconstruction of Array 2 (red markers—effect of lossy compression). We compute the SD from the 8 replicated arrays of the same type as Array 2. Since the SD bands are hard to distinguish in a plot, we show a standardized plot of z-statistics (divided by 2) in Figure 10.5. The method-to-method variability is much greater than the array-to-array and lossy-lossless differences, as indicated by the blue markers with large z-statistics. The lossy-lossless differences (red markers) are smaller than both the method-to-method (blue markers) and array-to-array (green markers) differences. The conclusions we can draw from this is that great care has to be taken in which method is used for information extraction, and furthermore that at bit rates close to 4 bpp the distortion introduced by the compression is below the replicate experiment variability. We obtain similar results for all other arrays, and when the roles of Method 1 and Method 2 are reversed.

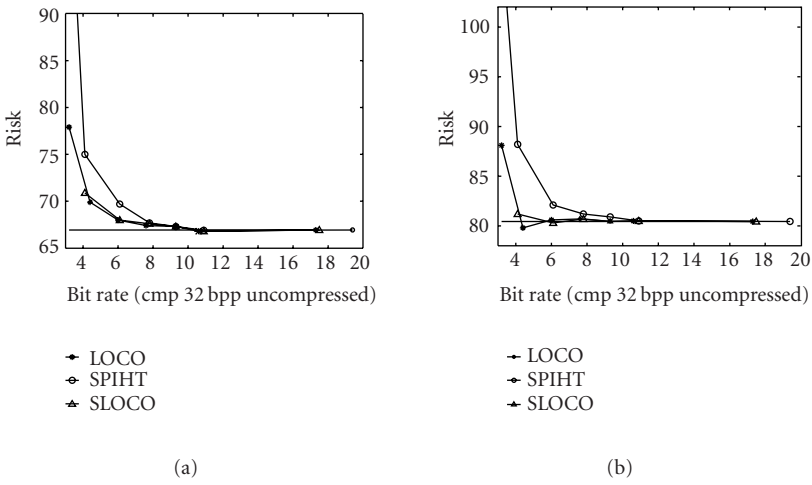


Figure 10.6. Method 1. (a) Control group: risk (summed over all genes on array) versus bit rate. (b) Treatment group: note that SPIHT risk exceeds the risk of LOCO and SLOCO for bit rates below 12 bpp. For bit rates above 6 bpp, we see only moderate increases in risk using lossy reconstruction via LOCO or SLOCO and a modest decrease in risk on the treatment array for 4 bpp.

10.4.4. Risk, denoising, and shrinkage

The replicated experiment allows us to construct a "ground truth" for the gene expressions. We compute the mean gene expression vector from the 8 replicates using Method 1, and denote this by M^0. We construct a measure of risk for the compressed data at various bit rates as follows. Let $\widehat{M}^j$ the be vector of extracted gene expressions from a lossy reconstruction or array j, and $\widehat{M}^j_k$ the kth element of this vector, that is, the differential gene expression of gene k on array j. If there are p genes on each array, the risk R is computed as

$$R = \sum_{k=1}^{p} \frac{1}{N_{\text{array}}} \left| \widehat{M}^j_k - M^0_k \right|. \tag{10.10}$$

Here, we have 8 replicate experiments and thus $N_{\text{array}} = 8$. We use the L_1 norm to avoid having the large gene expressions dominate the calculated risk.

In Figures 10.6a and 10.6b the risk, using Method 1, is plotted as a function of bit rate, using the SPIHT, LOCO, and SLOCO algorithms. The minimum decodable bit rate for the SLOCO is 4.1 bpp. The standard LOCO results are computed using LOCO at each bit rate separately, since LOCO is not a progressive scheme. SPIHT risk exceeds the risk of LOCO and SLOCO at all bit rates below 12 bpp, after which the three algorithms perform similarly. The SLOCO and standard LOCO show comparable risk results. However, there is a caveat in this comparison. We know that standard LOCO leads to a considerable bias in the estimate of A for low

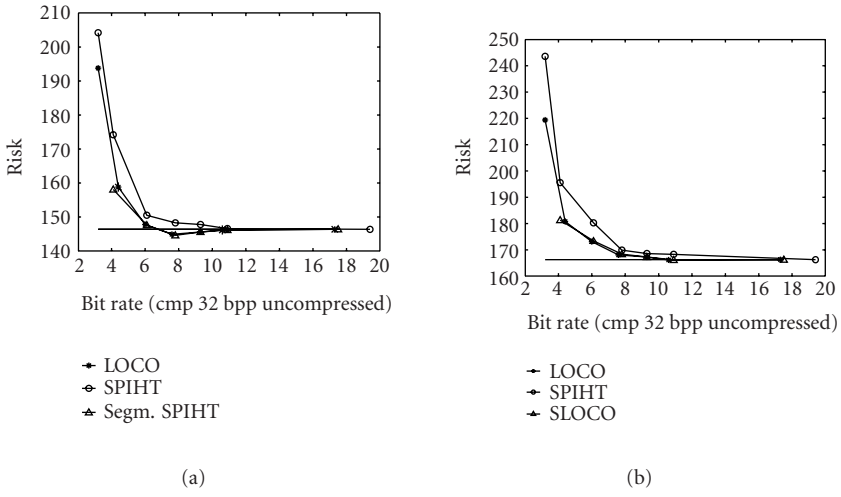


Figure 10.7. Method 2. (a) Control group: risk (summed over all genes on array) versus bit rate. (b) Treatment group.

bit rates, which Figure 10.6 does not show. The overall risk for the control group is smaller than the treatment groups (Figure 10.6a versus 10.6b), as the scale on the y-axes in the two figures indicate. This is due to the larger (absolute) gene expressions levels extracted from the treatment arrays compared to the control arrays. Both standard LOCO and SLOCO improves on the risk of the full data (horizontal line in figures) for bit rates ~6–8 bpp, for the treatment arrays (b). We can think of the compression at these bit rates as *denoising* of the microarray images. For both groups of arrays, we see only a marginal increase in risk for rates greater than 4 bpp.

In Figures 10.7a and 10.7b, we plot the risk using Method 2. The overall risk is much higher using Method 2 than using Method 1, as indicated by the scale of the y-axes in the figures. Method 2 introduces a lot of variability in the extracted information. We see again a denoising effect for part of the bit rate range in Figure 10.7a. For the entire bit rate range displayed, both standard LOCO and SLOCO show only marginal increase in risk over the risk obtained using the full data.

The equivalence of compression and denoising has been previously discussed in the literature (e.g., [18]). If the noise level is lower than the "signal," that is, the extracted gene expression, we can get a better estimate of the gene expressions using compressed data, than that we get using the full data. This is what Figures 10.6 and 10.7 illustrate. Moreover, the effect is more apparent when the method of extraction is susceptible to noise (Method 2). In Figure 10.8, the shrinkage, or denoising, effects of the compression on the extracted gene expressions are shown. The solid line is the mean gene expression vector M^0. The black dots correspond to the estimate of the gene expressions using a lossless reconstruction of Array 2.

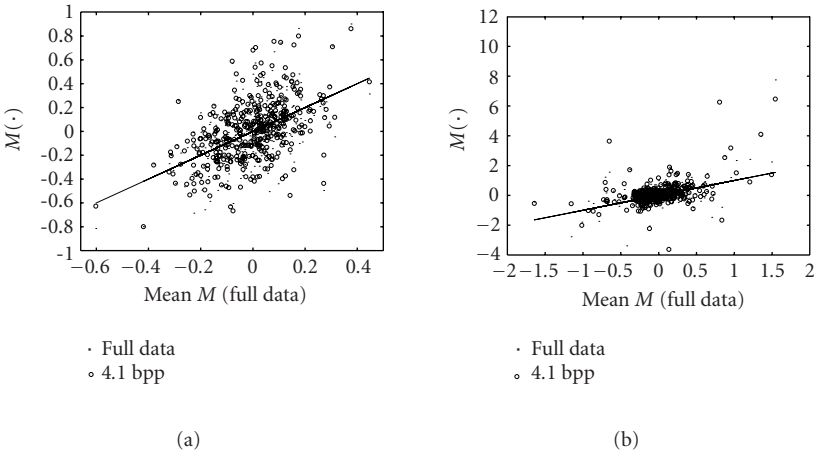


Figure 10.8. The equivalence of compression and densoising. The solid line depicts the mean M gene expression level across the replicate arrays. The dots $(\cdot)$ represent M (Array 2, full data), that is, the extracted gene expressions from one array only. The open circles $(\circ)$ represent M (Array 2, 4.1 bpp), that is, the extracted gene expressions from a lossy reconstruction of Array 2. In figure (a), Method 1 is used. In figure (b), Method 2 is used. Note that for large absolute M values compression leads to shrinkage of the single array estimates towards the mean M across the 8 arrays.

The open circles are obtained from a lossy reconstruction. We see that for large absolute values of M^0, the lossy reconstruction $\widehat{M}$ is closer to M^0 than the lossless reconstruction M. For the small and highly variable M^0, compression, or denoising, does neither good nor harm. If there is no significant signal present, we have no chance of retrieving it, no matter which method we use.

10.5. Discussion

We have presented a lossless and progressive compression scheme for microarray images. At bit rate 4.1 bpp, we find that the tasks of image segmentation, and genetic information extraction with a variety of methods, are only marginally affected by the compression. The effect is smaller than the array-to-array variability. The effect of compression is also smaller than the difference between alternative methods for information extraction.

Since our scheme is partially progressive, experimentalists can opt to refine the precision of the images. The flexible structure of our scheme allows for lossless, or refined, precision reconstruction for any subset of the images. Our scheme is robust to specific microarray image characteristics, and has been tried on images from three different labs.

We find that compression can in fact improve the estimation of gene expression levels. Using replicated arrays, we show that compression acts as a form of shrinkage for large absolute gene expression, toward the mean over the replicated arrays.

For simplicity, we have opted to code the two pairs of microarray images separately. However, a joint coding scheme may improve not only the lossless bit rate, but the lossy performance as well. Near low intensity spots, lossy reconstruction of the background region can have huge impact on the extracted information. We control this by using higher precision in these regions. A joint coding scheme can control the variance introduced by the compression by ensuring that the *joint* loss is smaller than a specific value. The worst case scenario in separate coding is that maximum errors of opposite sign are obtained in the two scans. This can be avoided with a joint coding scheme.

Even though we believe the multilayer data structure should work for emerging array technologies, such as the protein array, modifications specific to the makeup of a particular array and the application of the array in clinical diagnostic situations might be required to overcome challenges not seen in the cDNA microarray case. For example, from articles in [1], it is not hard to imagine that the antibody array technology would be one day applied in physician's offices, and an online version of the multilayer data structure would be needed and statistical analysis or expression level extraction should be based on compressed objects to speed up the computation or the waiting time of the patient.

Array technology is only one of the many imaging or recording technologies which provide challenges and new opportunities for the compression or signal processing community. More and more, we will be seeing the need to integrate compression and other signal processing tasks with data or statistical analysis to follow.

Bibliography

[1] C. A. K. Borrebaeck and C. Wingren, Eds., *A Supplement to BioTechniques: High-Thoughput Proteomics: Protein Arrays I*, Eaton Publishing, Westborough, Mass, USA, 2002.

[2] J. Hua, Z. Liu, Z. Xiong, Q. Wu, and K. R. Castleman, "Microarray BASICA: background adjustment, segmentation, image compression and analysis of microarray images," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 92–107, 2004, Special issue on Genomic Signal Processing.

[3] R. Jornsten, *Data compression and its statistiscal implications: with an application to the analysis of microarray images*, Ph.D. thesis, Department of Statistics, UC Berkeley, 2001.

[4] R. Jornsten, W. Wang, B. Yu, and K. Ramchandran, "Microarray image compression: SLOCO and the effect of information loss," *Signal Process.*, vol. 83, no. 4, pp. 859–869, 2003.

[5] Y. H. Yang, S. Dudoit, P. Luu, et al., "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation," *Nucleic Acids Res.*, vol. 30, no. 4, pp. e15, 2002.

[6] S. Dudoit, Y. H. Yang, M. J. Callow, and T. P. Speed, "Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments," *Statist. Sinica*, vol. 12, no. 1, pp. 111–139, 2002.

[7] Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed, "Comparison of methods for image analysis on cDNA microarray data," *J. Comput. Graph. Statist.*, vol. 11, no. 1, pp. 108–136, 2002.

[8] N. Merhav, G. Seroussi, and M. Weinberger, "Modeling and low-complexity adaptive coding for image prediction residuals," in *Proc. IEEE International Conference on Image Processing (ICIP '96)*, pp. 353–356, Lausanne, Switzerland, September 1996.

[9] Axon Instruments, GenePix 4000A, User's Guide 1999.

[10] M. Sapir and G. A. Churchill, *Estimating the posterior probabilities of differential gene expressions from microarray data*, The Jackson Laboratory, 2000, http://www.jax.org/staff/churchill/labsite/.

[11] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarrays images," *J. Biomed. Opt.*, vol. 2, pp. 364–374, 1997.
[12] M. Newton, C. Kendziorski, C. Richmond, F. Blattner, and K. Tsui, "On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data," Tech. Rep. 139, Department of Biostatistics and Medical Informatics, UW Madison, 1999.
[13] A. Said and W. Perlman, "Reversible image compression via multiresolution representation and predictive coding," *Proc. SPIE*, vol. 2094, pp. 664–674, 1993.
[14] S. M. Perlmutter, P. C. Cosman, C.-W. Tseng, et al., "Medical image compression and vector quantization," *Statist. Sci.*, vol. 13, no. 1, pp. 30–53, 1998.
[15] J. Ström and P. Cosman, "Medical image compression with lossless regions of interest," *Signal Process.*, vol. 59, no. 2, pp. 155–171, 1997.
[16] G. J. Sullivan, "Efficient scalar quantization of exponential and Laplacian random variables," *IEEE Trans. Inform. Theory*, vol. 42, no. 5, pp. 1365–1374, 1996.
[17] C.-C. Lu and J. G. Dunham, "Highly efficient coding schemes for contour lines based on chain code representations," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1511–1514, 1991.
[18] M. Hansen and B. Yu, "Wavelet thresholding via MDL for natural images," *IEEE Trans. Inform. Theory*, vol. 46, no. 5, pp. 1778–1788, 2000, Special issue on Information Theoretic Imaging.
[19] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 6, pp. 641–647, 1994.
[20] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000.
[21] T. Nakachi, T. Fujii, and J. Suzuki, "Pel adaptive predictive coding based on image segmentation for lossless compression," *IEICE Trans. Fundamentals*, vol. E82-A, no. 6, pp. 1037–1046, 1999.
[22] National Human Genome Research Institute report 1998, http://www.nhgri.nih.gov.
[23] D. Nister and C. Christopoulos, "Lossless region of interest coding," *Signal Process.*, vol. 78, no. 1, pp. 1–17, 1999.
[24] A. Ortega and K. Ramchandran, "From rate-distortion theory to commercial image and video compression technology," *IEEE Signal Processing Mag.*, vol. 15, no. 6, pp. 20–22, 1998.
[25] S. M. Perlmutter, P. C. Cosman, C.-W. Tseng, et al., "Medical image compression and vector quantization," *Statist. Sci.*, vol. 13, no. 1, pp. 30–53, 1998.
[26] L. Shen and R. M. Rangayyan, "A segmentation-based lossless image coding method for high-resolution medical image compression," *IEEE Trans. Med. Imag.*, vol. 16, no. 3, pp. 301–307, 1997.

Rebecka Jörnsten: Department of Statistics, Rutgers University, Piscataway, NJ 08855, USA
Email: rebecka@stat.rutgers.edu
Current address: Department of Statistics, University of California, Berkeley, CA 94720-3860, USA

Bin Yu: Department of Statistics, Rutgers University, Piscataway, NJ 08855, USA
Email: binyu@stat.berkeley.edu
Current address: Department of Statistics, University of California, Berkeley, CA 94720-3860, USA

11 Cancer genomics, proteomics, and clinic applications

X. Steve Fu, Chien-an A. Hu, Jie Chen,
Z. Jane Wang, and K. J. Ray Liu

Preface

Throughout the history of medicine, many advances are derived from important innovations in technology. For example, the invention of the X-Ray machine has revolutionized medicine and pioneered modern imaging. The invention of the microscope essentially redefined the field of pathology and microbiology. In the past few decades, "technology explosion" has created an immense impact on both biomedical research and clinical medicine. Tremendous strides were made with the aid of numerous new technologies such as recombinant DNA methods, DNA sequencing, magnetic resonance imaging (MRI), polymerase chain reaction (PCR), monoclonal antibodies, and so forth. Despite these, major hurdles remain. In the field of cancer medicine, limited successes are still overshadowed by the tremendous morbidity and mortality incurred by this devastating disease. It has become increasingly important to integrate new technologies into both cancer research and clinical practice if we hope to win the battle against cancer.

In this chapter, we will briefly review the molecular basis of cancer and our current understanding. We will focus our attention on genomics and proteomics of cancer. We believe that a thorough understanding of the DNA and protein compliments of cancers that dictate the subsequent disease phenotype would eventually lead to breakthroughs. The impact of modern technology on cancer diagnosis, prognosis, and treatment will also be discussed. We placed our emphasis on two of the cutting-edge technologies, microarray technology and nanotechnology, as they are clearly among the leading frontiers that will rapidly reshape biomedical sciences and clinical oncology. Finally, we will discuss our current active research to facilitate our understanding and management of cancer.

11.1. Understanding cancer

11.1.1. Overview

The financial and societal burden of common diseases such as cardiovascular, metabolic (e.g., diabetes), and neoplastic diseases (cancer) is very significant.

Table 11.1. Number of people affected and annual cost of common diseases in US population. (Data derived from Center of Disease Control (CDC); http://www.cdc.gov/nccdphp/major_accomplishments.htm).

Disease	Number of people affected (million)	Cost (billion)
Cardiovascular disease	58	$287
Arthritis	43	$65
Diabetes	16	$98
Cancer	8.4	$107
Alzheimer's disease	4	$152
Schizophrenia	2	$30
Osteoporosis	1.5 fracture/year	$14

Cancer is the fourth most common disease and the second leading cause of death in the United States, accounting for nearly one quarter of total human mortality. More than 500 000 people die from some forms of cancer each year in the US. The costs are not only borne by the patients and health-care system for medical expenses, but also caused by lost productivity and premature death. Examples of the magnitude of the costs are shown in Table 11.1. Therefore, beneficial and effective interventions in screening, diagnosis, prognosis, and treatment are desperately needed.

There have been significant discoveries over the past several years in our understanding of the genetic basis of human cancer. Colorectal cancer (CRC), for example, affected approximately 135 000 people in the US in 2001, resulting in approximately 57 000 deaths [1]. CRC develops as the result of the progressive accumulation of genetic and epigenetic alterations that lead to the transformation of normal colonic epithelium to adenocarcinoma. The loss of genomic stability is a key molecular and pathophysiological step in this process and serves to create a permissive environment for the occurrence of alterations in cancer-related genes (e.g., tumor-suppressor genes and oncogenes). Alterations in these genes (e.g., APC, K-RAS, and p53, as shown in Figure 11.1) appear to promote colon tumorigenesis by perturbing the function of signaling pathways or by affecting genes that regulate genomic stability [2]. Epigenetics is generally defined as a modification of the genome, inheritable by cell progeny, that does not involve a change in DNA sequence. For example, genomic imprinting, a form of epigenetic inheritance, is a modification of a specific parental allele of a gene, or the chromosome on which it resided, in the gamete or zygote leading to differential expression of the two alleles of the gene in somatic cells of the offspring. One mechanism of genomic imprinting is altered methylation of CpG islands, differentially methylated regions of a tumor-suppressor gene [3]. It is evident that environmental factors (e.g., smoking, diet, and exercise) can certainly contribute to a person's risk of cancer.

Again, using CRC as an example, it develops as the result of the progressive accumulation of genetic and epigenetic alterations that lead to the transformation of normal colonic epithelium to adenocarcinoma, as shown in Figure 11.1. The fact that CRC develops over 10–15 years and progresses through parallel histological

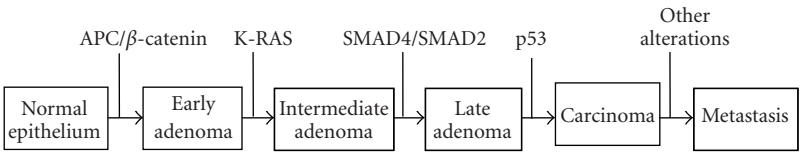


Figure 11.1. Genetic model of colon tumorigenesis.

and molecular changes has permitted the study of its molecular pathophysiology in more detail than other cancer types. From the analysis of the development of CRC, three key themes concerning the molecular pathogenesis of cancer have been established. First, cancer emerges via a multistep progression at both the molecular and the morphologic levels (refer to Figure 11.1). Second, loss of genomic stability is a key molecular and pathophysiological step in cancer formation. Third, hereditary cancer syndromes frequently correspond to germ-line forms of key genetic defects whose somatic occurrences drive the emergence of sporadic colon cancers [3]. We elaborate on these themes.

(1) *Sequential events of CRC tumorigenesis.* The evolution of normal epithelial cells to adenoma, and then to carcinoma usually follows a predictable progression of histological changes and concurrent genetic and epigenetic changes. These alterations provide a growth advantage and lead to clonal expansion of the altered cells. Subsequent alterations with waves of clonal expansion then occur as a consequence of progressive events that provide other growth advantages to the cells such as loss of cell contact inhibition. The earliest identifiable lesion in colon-cancer formation is the aberrant crypt focus (ACF). The true neoplastic potential of this lesion is still undetermined, but it does appear that some of these lesions can progress to adenocarcinoma and harbor mutations in associated genes (e.g., APC). Subsequent alterations in other genes then play a role in tumor growth and the eventual acquisition of other malignant characteristics such as tissue invasiveness and the ability to metastasize (refer to Figure 11.1).

(2) *Genetic alterations.* The p53 protein was initially identified as a protein that formed a complex with the SV40 large T-antigen and was originally suspected to be an oncogene [4]. Subsequent studies demonstrated that p53 is a transcription factor with tumor-suppressor activity. Human p53 structural gene is located at chromosome 17p13.1 and mutated in more than 50% of primary human tumors, including tumors of the gastrointestinal tract. p53 maintains genomic stability through the control of cell cycle progression and apoptosis (programmed cell death) in response to genotoxic stress (e.g., UV, toxic chemicals). In CRC, p53 mutations have not been observed in colon adenomas, but rather appear to be late events in the colon adenoma-carcinoma sequence that may mediate the transition from adenoma to carcinoma. Furthermore, mutation of p53 coupled with loss of heterozygosity (LOH) of the wild-type allele was found to coincide with the appearance of carcinoma in an adenoma, thus providing further evidence of its role in the transition to malignancy [2]. LOH, or allelic imbalance, is generally defined

as inactivation of the wild-type copy of a tumor-suppressor gene by deletion, gene conversion, mitotic recombination (rearrangement), or loss of an entire chromosome. p53 normally serves to regulate cell growth and division in the context of genotoxic stress. It is expressed at very low levels in cells until it is activated via incompletely understood mechanisms by DNA damage as a result of gamma irradiation, UV irradiation, or chemotherapeutic agents. Its activation results in the transcription of genes that directly regulate cell cycle progression and apoptosis. This function of p53 to recognize DNA damage and induce cell cycle arrest and DNA repair or apoptosis has led to p53 being called the "guardian of the genome." Thus, p53 normally acts as a tumor-suppressor gene by inducing genes that can cause cell cycle arrest or apoptosis and also by inhibiting angiogenesis (new blood vessel formation) [2].

(3) *Genomic instability.* In addition to having effects on cell biology, some genetic and epigenetic alterations do not directly affect the cell biology of the tumor, but instead result in the loss of genomic stability, which contributes to the accumulation of mutations in oncogenes (K-RAS) and tumor-suppressor genes (e.g., p53). The timing of the loss of genomic stability, either chromosomal instability or microsatellite instability, appears to be after adenoma formation but before progression to malignancy. One key concept that has emerged is that these alterations involve specific molecular pathways in colon cancer formation, and the alterations in these pathways presumably result in specific biological effects that promote carcinogenesis.

In summary, CRC genetics has yielded new insights and paradigms that have broadly informed the studies of most solid tumors. Key insights that have been contributed include the multistep nature of tumorigenesis, the central role of tumor-suppressor pathways, the key role in cancer of mutational inactivation of p53, and the role of DNA repair genes and genomic stability in cancer prevention. Nonetheless, many challenges remain. The genesis of the metastatic phenotype that directly accounts for cancer lethality still remains a new frontier awaiting further exploration. A mechanistic understanding of the basis of chromosomal instability of the cancer genome has yet to be achieved. Nor is there yet an understanding of the genetic basis within the general population of individual susceptibility to colon or other cancers. Lastly, the translation of molecular genetics to new diagnostic, prognostic, and therapeutic interventions remains a challenge. The identification of these alterations has provided targets for the development of new strategies for the prevention and/or treatment of colon tumors throughout their progression from normal epithelium to carcinoma. Indeed, pharmaceutical and biological agents that target alterations such as mutant p53 and oncogenic K-RAS are currently in clinical trials (refer to our later discussion).

11.1.2. Genomics, Human Genome Project, and postgenomic era

Over the past few years, life science-based research has witnessed revolutionary progress driven by high-throughput measurement technologies for biological

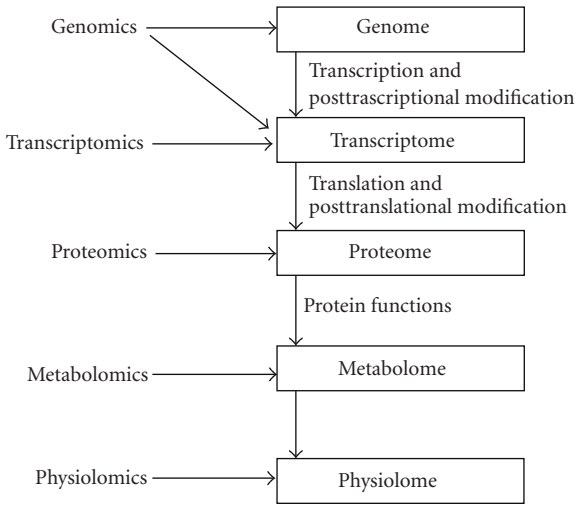


Figure 11.2. Schematic plot of what is included in system biology.

molecules, with a shift from an individual approach (genomics, proteomics, etc.) towards an integrated approach. The new integrated approach naturally led to the emerging field of *systems biology* focusing on achieving a system-level understanding of a biological unit, as shown in Figure 11.2. Next, we will give a brief overview.

11.1.2.1. Genomics

Genome, a term coined by Winkler in 1920 as a conjunction between gene and chromosome, symbolizes the haploid chromosome set together with its inclusive genes [5]. The Human Genome Project (HGP) with its principal goals of completely mapping and sequencing the human genome was conceived in 1984 and implemented in 1990. It was principally completed in 2003 [6, 7, 8]. The "final" completion of HGP will be the sum total of a number of genomic studies; namely, (a) *the physical genome*: the gene map, control motif map, and full DNA sequence; (b) *the functional and structural genome*: an understanding of how genes are organized and what they do; (c) *the population genome*: the variation of genes in the human population; (d) *the comparative genome*: the comparison of the human genome with other genomes; and (e) *the integrative genome*: the functional interaction of genes within the genome, among others to be invented in the future.

In this postgenomic era, researchers will access, analyze, and mine vast volumes and multiple types of data on a grand scale. New "computer-based approaches" or "algorithm-based approaches" have appeared to tackle postgenomic challenges such as functional genomics, comparative genomics, proteomics, metabolomics, pathway analysis, and systems biology.

Functional genomics has emerged recently as a new discipline employing major innovative technologies for genome-wide analysis supported by information

technology. The widely used term "functional genomics" has many different interpretations. In one possible definition, functional genomics refers to the development and application of global (genome-wide or system-wide) experimental approaches to assess gene function by making use of the information and reagents provided by genome sequencing and mapping [9]. It is also defined as the branch of genomics that determines the biological function of the genes and their products, where this function can be defined at several levels, including biochemical, cell biology tissue, organ, and organismal. Due to the function at different levels, we can identify a set of "vertical" areas such as DNA arrays for expression profiling, proteomics, structural genomics, high-throughput description of cellular systems, and so forth.

Functional genomics is characterized by large-scale, massively parallel experimental methodologies (generating data on gene expression, protein structure, protein interactions, etc.) combined with statistical and computational analysis of the results, and by mining the results for particularly valuable information. The fundamental strategy in a functional genomics approach is to expand the scope of biological investigation from studying single genes or proteins to studying all genes or proteins at once in a systematic fashion. Functional genomics promises to rapidly narrow the gap between sequence and function and to yield new insights into the behavior of biological systems.

There are many limitations of gene-expression profiling using genomics approaches. Previously, systematic investigation of gene expression using different genomic methodologies such as subtractive hybridization, differential display, serial analysis of gene expression (SAGE), and expression microarray have generated some interesting and useful datasets. Although a systematic analysis of mRNA expression can provide a profile of a cell/tissue transcriptome, there may be marked discrepancies between mRNAs and protein abundances [10]. Many changes in gene expression might not be reflected at the level of protein expression or function. Moreover, quantitative mRNA level is insufficient to predict actual protein expression because of posttranscriptional regulation and internal ribosome initiation of translation [11, 12]. Finally, proteins are subjected to posttranslational modifications and their stability (turnover rate) is regulated under varying physiological conditions. Therefore, proteomics is an emerging field to the discovery and characterization of regulated proteins or biomarkers in different diseases in the postgenomic era.

11.1.2.2. Proteomics

Proteomics is the study of the protein complement of a cell. It is the proteome and the collective functions of proteins that directly dictates the phenotype of the cell. Proteomics strives to profile and characterize the entire proteome or a specific component (subproteome) [13, 14, 15]. In recent years, protein separation methods coupled with various mass spectrometry (MS) technologies have evolved as the dominant tools in the field of proteomics [16]. Technically, this is not trivial and requires a number of sophisticated techniques in combination that involve

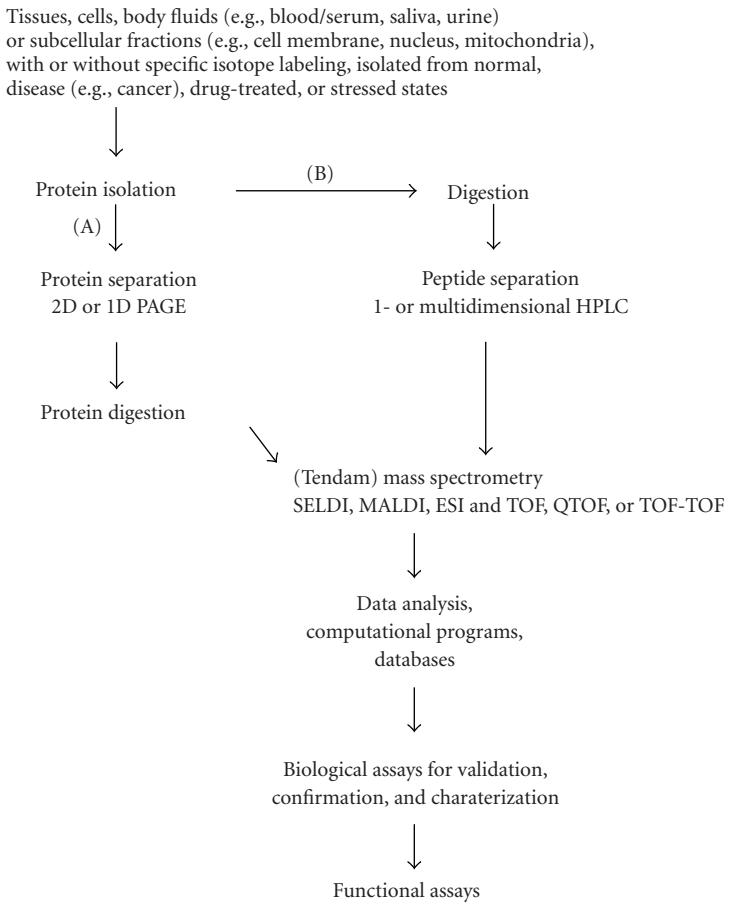


Figure 11.3. Schematic presentation of proteomic methodologies.

sample identification and isolation, protein separation, identification by MS, and functional characterization, as shown in Figure 11.3. The brief description of each step is as follows.

Sample identification and isolation. It can be as straightforward as obtaining a tissue biopsy or body fluids (e.g., blood/serum, saliva, urine) from an individual, or it may involve the precise isolation of a cell or cluster of cells from a biopsy specimen by using different micromanipulation (e.g., fluorescence-activated cell sorter (FACS), laser capture microdissection). To increase both specificity and sensitivity, acquired samples can be labeled in advance with radio-isotope precursors. Once the sample has been obtained, it is subjected to protein isolation by removing DNA, RNA, carbohydrates, and lipids. Typically, this step is accomplished by organic solvent (e.g., methanol) extraction. After protein isolation, two different

tracks (track (A) and track (B) as indicated in Figure 11.3) can be used to generate peptides for MS analysis as follows.

(i) **Track (A)—direct protein separation**. Traditionally, the separation of extracted proteins is accomplished by 1-dimensional (1D) or 2D polyacrylamide gel electrophoresis (PAGE). Using 2D PAGE as an example, in the first dimension, proteins are separated according to their isoelectric point (or net charge) in a tube gel; whereas in the second dimension, proteins are separated according to their molecular masses in the orthogonal direction using electrophoresis in a slab gel containing sodium dodecyl sulfate. After gel staining, interested protein spots that show altered intensity compared to the control are cut out and subjected to protease (typically trypsin) digestion. When first developed, 2D PAGE was believed to provide unique and unequivocal protein separation with each "spot" on the gel corresponding to a single protein. Using this approach, several thousand protein spots can be resolved in a single slab gel. However, subsequent analysis using highly sensitive MS techniques has shown that this view is incorrect: many, if not most, spots on a 2D PAGE contain multiple proteins. Furthermore, sensitivity of 2D PAGE is also limited by the staining method used to detect protein spots on the gel. Thus, due to the lack of sensitivity and specificity, alternative methods (see Track (B)) have been developed recently.

(ii) **Track (B)—protein digestion and separation**. The coupling of liquid chromatography (LC) with MS has had a great impact on small molecule and protein profiling, and has proven to be an important alternative to 2D PAGE. Typically, proteins are first digested by protease into small peptides then the derived peptide mixture are separated by ionic or reverse phase LC and subjected to MS analysis. LC-MS has been applied to large-scale protein characterization and identification. In addition to its role in protein profiling, LC-MS is perhaps the most powerful technique for monitoring, characterization, and identification of impurities in pharmaceuticals. Because of the complexity of any given proteome and the separation limits of both 2D PAGE and LC, only a fraction of that proteome can be analyzed. An alternative approach is to reduce the complexity prior to protein separation and characterization. Many of these approaches involve LC methods, which utilize solid- and liquid-phase media to separate proteins according to specific biochemical properties (e.g., molecular mass, isoelectric point, hydrophobicity). These LC separations can be performed in series (or multidimensional) to improve resolving power. Furthermore, if one is interested in a specific class of proteins (e.g., those bearing phosphate group(s)) or proteins with posttranslational modification (e.g., glycosylation), unique columns that contain a matrix specific for these functional groups can be used to separate these proteins from all others by affinity chromatography. For many of the LC approaches, proteins are subjected to proteolytic digestion to afford a multitude of peptides derived from each protein.

Protein identification by MS. Once separated by either track, the resulting peptides require identification. Currently available methods all use some form of MS. MS is a rapidly evolving methodology that converts proteins or peptides to charged species that can be separated on the basis of their mass-to-charge ratio (m/z).

These methods have been considered a major advance in the identification of polypeptides from the proteome. John Fenn and Koichi Tanaka shared the Nobel Prize in chemistry in 2002 for their contributions to the MS field [17]. MS requires that proteins or peptides are first converted to gas-phase ions within a specific region of the instrument, the ionization source. The ions are then separated with a mass analyzer on the basis of their m/z. The resulting mass spectra are represented as plots of intensity versus m/z. There are several different types of MS ionization methods currently available, including surface-enhanced laser desorption ionization (SELDI), electrospray ionization (ESI), and matrix-assisted laser desorption ionization (MALDI), which have made it possible to analyze large biomolecules, such as peptides and proteins. Another key developments were the invention of the time-of-flight (TOF) MS and relatively nondestructive methods to convert proteins into volatile ions. Using MALDI-TOF and ESI-TOF as examples, essentially, the resulting charged peptides that are detected in MALDI or ESI can next be subjected to high-speed collision with an inert gas, such as argon, yielding smaller charged fragments that can be pieced together to reconstruct peptide sequences. Peptide sequences identified with these methods must next be analyzed by comparison with known database sequences to determine the unequivocal identity of the protein. When MALDI is used, the peptide samples are solidified within an acidified matrix, which absorbs energy in a specific UV range and dissipates the energy thermally. This rapidly transferred energy generates a vaporized cloud, thereby simultaneously ejects the analytes into the gas phase where they acquire charge. A strong electrical field between the MALDI plate and the entrance of the MS tube forces the charged analytes to rapidly reach the entrance at different speeds and times based on their m/z ratios. A significant advantage of MALDI-TOF is that it is relatively easy to perform protein or peptide identification with moderate throughput (96 samples at a time). MALDI-MS provides a rapid way to identify proteins when a fully decoded genome is available because the deduced masses of the resolved analytes can be compared to those calculated for the predicted products of all of the genes in the human genome.

The ESI method is also widely used to introduce the mixtures of peptides into the biological MS instrument. The unique feature of ESI is that at atmospheric pressure it allows the rapid transfer of analytes from the liquid phase to the gas phase. The spray device creates droplets, which once in the MS go through a repetitive process of solvent evaporation until the solvent has disappeared and charged analytes are left in the gas phase. Compared with MALDI, ESI has a significant advantage in the ease of coupling-to-separation techniques such as LC and high-pressure LC (HPLC), allowing high throughput and online analysis of peptide or protein mixtures.

Currently, proteomics analysis is rapidly switching to MALDI-TOF or ESI-TOF coupled with tandem MS, that is, MS/MS. Typically, a mixture of proteins is first separated by LC followed by MS/MS. In this procedure, a mixture of charged peptides is separated in the first MS according to their m/z ratios to create a list of the most intense peptide peaks. In the second MS analysis, the instrument is adjusted so that only a specific m/z species is directed into a collision cell to generate

"daughter" ions derived from the "parent" species. Using the appropriate collision energy, fragmentation occurs predominantly at the peptide bonds such that a ladder of fragments, each of which differs by the mass of a single amino acid, is generated. The daughter fragments are separated according to their m/z, and the sequence of the peptide can then be predicted by comparing the databases.

Radioisotope labeling-assist proteomics. Once proteins in a given proteome have been identified, their relative abundance levels need to be determined, especially if one purpose of the experiment is to determine the comparative abundance of a protein or proteins in normal versus diseased states. Without using 2D PAGE, direct protein quantization based on MS signals remains a challenge because of the nonlinear correlation between protein quantity and MS ionization efficiency [13, 18]. Therefore, a number of isotope-tagging methods such as isotope-coded affinity tag (ICAT), have been introduced for providing MS recognizable markers/references in protein quantification [19]. To determine relative species abundance, the ICAT method uses a pair of reagents, containing a biotin moiety and a linker chain with either eight deuterium or eight hydrogen atoms, to differentially label protein samples on their cysteine residues. Two samples, each labeled with the ICAT reagent carrying one of the two different isotopes, were mixed and subjected to site-specific protease digestion. The labeled peptides containing cystine can be highly enriched by binding the biotin tags to streptavidin, resulting in a greatly simplified peptide mixture. Characterization of the peptide mixture was carried out by the LC-MS approach as described above. Quantization of differential protein expression level can be achieved by comparing the areas under the doublet peaks that are separated by eight mass units. Thus, the ICAT method works well for the differential analysis of many proteins in a complex mixture: one of natural abundance and the other isotopically labeled, quantitative differences in abundance of proteins between the two different samples can be readily determined. The obvious limitation of the ICAT labeling approach is that a protein has to contain at least one cysteine residue to be detected [14]. Another labeling strategy, amino-acid-coded mass tagging (AACT) with stable isotopes through in vivo culturing [15], provided an accurate and comprehensive approach for quantitative protein analysis on a proteome-scale [20]. For example, we have utilized the AACT strategy coupled with both LC-MS/MS and MALDI-TOF MS to profile global protein expression in p53-induced apoptosis in a human CRC cell line that harbors an inducible p53 gene (GU, Molecular and Cellular Proteomics, in press).

Proteomics in cancer research. In cancer, the proteome of the cancer cells experience continuous and dynamic alterations [16]. Individual protein changes can be functionally benign, synergistic, or detrimental, but collectively, they can result in malignant phenotypes. However, the changes to the proteome can also be protective, resulting in systemic stress responses. In general, cancer cells have defects in the regulation of either cell cycle or apoptosis (programmed cell death), enabling them to gain survival advantages and immortality. Studies are designed to uncover the molecular bases of cancer development and to restore functional cell cycle

arrest and apoptosis in cancer cells. Proteomics has been used in understanding cell cycle and apoptosis. Using apoptosis as an example, this process is an essential and highly regulated physiological function required for normal development and maintenance of tissue homeostasis in all multicellular organisms. The function of this process is to eliminate unwanted or injured cells with characteristic cellular and biochemical hallmarks. Dysregulation of apoptosis is evident in many human diseases including cancer, neurodegenerative disorders, and AIDS. p53, a tumor-suppressor protein and a transactivating factor, plays a pivotal role in regulating cell cycle arrest, differentiation, and apoptosis. The elevated expression of p53 leads to mitochondrial-mediated apoptosis. p53-induced apoptosis plays a critical role in the suppression of tumorigenesis. LOH and mutations in p53 were found in more than 50% of all human tumors. Previously, systematic investigation of p53-induced apoptosis has been explored by four different genomic methodologies, namely SAGE, microarray, differential display, and subtractive hybridization. Approximate 150 genes have been shown to be transcriptionally upregulated by p53 [10]. A proteomic study compared the proteome of a human CRC cell transfected with inducible p53 (DLD-1.p53) with that of the control DLD-1.vector cell line (Gu, Molecular a Cellular Proteomics, online publication). Using AACT-assisted MS, the group has systematically identified those proteins upregulated by the p53-mediated apoptosis. In cell culturing, the deuterium-labeled (heavy) amino acids were incorporated *in vivo* into the proteome of the DLD-1.p53 cells at the time of p53 induction, whereas the DLD-1.vector cells were grown in the unlabeled medium. In high throughput LC-ESI MS/MS analyses, the AACT-containing peptides derived from the mixture of equal numbers of DLD-1 and DLD-1.p53 cells were paired and their relative isotopic peak intensities gave the relative differential protein expression levels. In response to p53 overexpression, those proteins that changed their level of expression were found to be associated with six distinct function categories: cell cycle arrest and p53 binding, protein chaperoning, plasma membrane dynamics, stress response, antioxidant enzymes, and anaerobic glycolysis. This quantitative proteomic dataset suggests that the p53-induced apoptosis involves the activation of multiple pathways that are glycolysis relevant, energy dependent, oxidative stress mediated, and possibly mediated through interorganelle crosstalks. This profiling provides a new insight into a global view of p53-induced apoptosis, as shown in Figure 11.4.

Serum-based biomarkers identified through clinical proteomics. Clinical proteomics is the application of proteomics specific to the field of medicine. It encompasses the translation of proteomic technologies and methods into the production of diagnostics and therapeutics for the direct application to human health. Instead of using cancer tissues or cells, development on proteomic strategy for analysis of body fluids (e.g., serum, urine, and saliva) for biomarkers is also pivotal [21]. For example, tracking and monitoring the disease-induced modifications and increase of prostate-specific antigen (PSA) in serum of prostate-cancer patients, is a gold standard for diagnosis of prostate-cancer progression. Another example is a revolutionary approach for the early diagnosis of ovarian cancer using SELDI-MS.

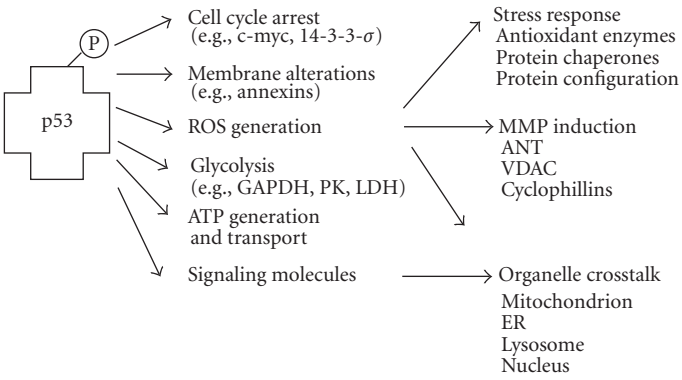


Figure 11.4. Functions of p53 in apoptosis. p53-regulated proteins and pathways in apoptosis were identified by genomic and proteomic approaches and can be grouped into five major "intermediate" functions; cell cycle arrest, membrane alterations, reactive oxygen species (ROS) generation, glycolysis, ATP generation and transport, and signaling; and three major "downstream" functions; stress response, mitochondrial membrane permeability (MMP) induction, and organelle crosstalk. Some representative proteins of each group are shown in parenthesis.

A clinical proteomics program jointly run by the NCI and the FDA developed a diagnostic test that is 100% sensitive and 95% specific in detecting ovarian cancer. Rather than looking for a specific protein, the group analyzed blood/serum samples for multiple protein biomarkers and found a "cancer signature." In other words, this proteomic pattern analysis relies on the pattern of proteins observed and does not rely on the identification of a traceable biomarker. Because of the high-throughput nature of this technique, hundreds of clinical samples per day can be analyzed utilizing this technology, which has the potential to be a novel, highly sensitive diagnostic tool for the early detection of cancer. Nevertheless, some scientists have reservations on using serum directly because serum is a complex, ever changing source of proteins. A marker found there may not be traced to organs easily, and the marker might be a body's systemic response to cancer instead of a specific signal derived from the tumor itself.

In summary, proteomics, the systematic evaluation of changes in the protein constituency of a cell, is more than just the generation of lists of regulated proteins that increase or decrease in expression as a cause of physiological change or consequence of disease. The ultimate goal is to characterize the information flow through protein pathways that interconnect the extracellular microenvironment with the control of gene transcription. The nature of this information can be a cause or a consequence of disease processes. Applications of proteomics to cancer research are limited at the current time, but are rapidly evolving. Cancer biologists have made the first attempts to utilize proteomics for diagnostic and prognostic purposes. Studies of proteomic patterns in sera of patients with breast or ovarian cancer provide examples of this strategy. One need not know the function of a protein when using this approach; rather, proteomic profiling using sera offer the possibility of identifying simple associations for diagnosis,

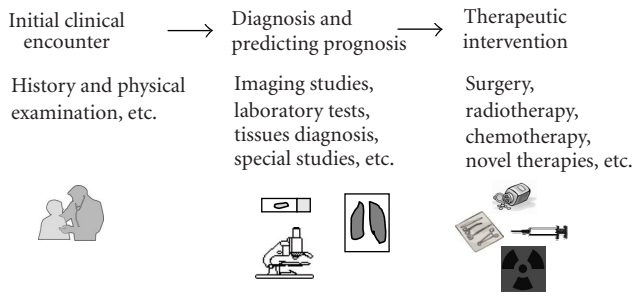


Figure 11.5. Typical ways of cancer diagnosis, prognosis, and treatment.

prognosis, and response to therapies. Clinical proteomics involve the use of proteomic methodologies at the bedside. The analysis of human cancer as a model for how proteomics can have an impact at the bedside is now employing several new proteomic technologies that are being developed for early detection, therapeutic targeting and patient-tailored therapy. Additionally, proteomics pushes technical limits as it attempts to provide information to complete our understanding of how the cell, tissue, organ or in some cases the whole system adapts during diseases. MS-based approaches have emerged as powerful tools for the genome-wide profiling of cellular or organismal proteins. However, proteomic analysis is currently limited by sensitivity, specificity, and throughput. Sensitivity is rapidly improving, with detection at the attomole (10^{-18} mol) level achieved by current MS methods, although this benchmark is not yet a routine. Specificity continues to improve, especially with application of multidimensional LC methods in place of 2D PAGE, and looking for patterns of biomarkers rather than single species. Finally, high throughput remains a challenge in proteomics analysis, however, some newer devices are designed to accommodate multiplexing of samples.

11.2. Advances in cancer diagnosis and prognosis

It is well known that the accumulation of genetic changes within cells could lead to cancer, and the activation of many genes are critical to the process of carcinogenesis. Therefore, an understanding of these genetic changes would be essential for cancer prevention, diagnosis, prognosis, and treatment. This also poses one of the key challenges now facing the cancer community. Although there is currently no cure for most forms of cancer, many options including surgery, chemotherapy, radiotherapy, and others novel approaches can be effective in prolonging survival, preventing metastasis, and improving patients' quality of life. Traditional ways of diagnosing cancer and predicting prognosis involve clinical examination in combination with pathologic evaluation (examining features of cancer under microscope), laboratory testing (blood works, etc.), and imaging studies (X-rays, etc.), as shown in Figure 11.5.

While clinical examination will always be one of the most critical and indispensable parts of establishing cancer diagnosis and prognosis, emerging new technologies have significantly improved its accuracy. This is especially true with cellular/molecular tests and the newer generations of radiographic imaging studies. In this section of the chapter, we will discuss some of the important advances in technology that can be applied to cancer diagnosis and prognosis. We will first review some of the current technologies in laboratory testings, radiographic imaging, and nuclear medicine briefly. We will then focus on using microarray technology for cancer diagnosis and prognosis, a leading technology that is rapid and radically changing the landscapes of cancer research and clinical practices.

11.2.1. Current techniques in cancer diagnosis and prognosis

11.2.1.1. Radiographic imaging and nuclear medicine

Various imaging studies are integral parts of a cancer workup. Over the past two decades, cancer imaging has relied heavily on cross-sectional studies. Computerized tomography (CT) is the single most used imaging modality for this purpose. Identification of disease processes is largely dependent upon the degree of structural changes. While this strategy is still widely used in practice, there are obvious limitations. For example, cancers may not always be obvious anatomically and therefore can be missed. Similarly, viable cancer residuals and scar tissues left behind after treatment cannot always be distinguished accurately just by the structural image. In the past decade, many new imaging techniques are emerging that have crossed beyond the traditional boundaries. The trend is shifting from a pure anatomical evaluation toward incorporating metabolic information into imaging studies (also referred to as functional imaging). To illustrate this, we discuss some of these new imaging modalities. Positron emission tomography (PET) imaging is a form of nuclear imaging that is extremely useful in cancer management [22]. PET takes advantage of the fact that cancer cells have higher metabolic rates and take up greater amounts of glucose than surrounding normal tissues. It employs an analog of glucose, [18F]-2-fluoro-2-deoxy-d-glucose (FDG). FDG enters into tumor cells via glucose transporter and is phosphorylated by intracellular enzymes. Unlike regular glucose, FDG cannot be metabolized by glycolytic enzymes in the cells, and is "trapped" in the cells. The accumulation of FDG in tumor cells therefore allows the detection of tumors as high intensity signals. In contrast, scar tissue and some other benign lesions are not FDG avid (the intensity of the signal is usually similar to that of the background) and therefore can be distinguished from malignant tumors. Magnetic resonance spectroscopy (MRS) is another example. Using existing magnetic resonance technology, it can analyze the chemical composition in an area of interest, such as a lesion, to assist in diagnosis [23]. There are three important spectroscopic signals: N-acetylaspartate, choline, and lactate. By analyzing the signal patterns detected with MR spectroscopy, physicians can narrow down the differential diagnoses and select their clinical approaches. Currently, MR spectroscopy is increasingly utilized as a diagnostic technique for brain

tumors. This technique may improve the differentiation of locally infiltrative brain tumors from other types of benign, well-circumscribed intracranial lesions. It can also help differentiate between neoplasm and other central nervous system processes such as infections or necrosis. Potential application of MRS in other types of malignancies includes head and neck cancer and lymphoma. Areas of research for applications of MRS in nonmalignant diseases such as strokes and dementia are also ongoing. It is not only worth mentioning that some of these new imaging modalities can be used for detecting and evaluating cancer, but can also be used to monitor drug penetration and distribution in solid organs or tissues.

11.2.1.2. Laboratory tests using cellular and molecular tools

Among many approaches in assessing a patient's illness, including that of cancer patients, one of the initial evaluations invariably involves obtaining some form of laboratory data. This can be as easy as a simple blood draw, or can be more complicated such as obtaining a bone marrow biopsy. For several decades, physicians and scientists were trying to identify specific "tumor markers" in patients' blood so as to help diagnose and prognose cancer. Despite limited success in doing so, a majority of tumors lack specific markers. Up to now, only a handful of tumor markers are routinely used, for example, alpha fetoprotein (AFP) for hepatocellular carcinoma and germ cell tumor, beta subunit of human chorionic gonadotropin (beta-HCG) for choriocarcinoma, PSA for prostate cancer, and CA 125 for ovarian cancer, and so forth. Despite their wide clinical application, the lack of sensitivity and specificity limit their usefulness. More recently, newer generations of laboratory testing are becoming available as a result of new technologies. Several examples, including the PCR and flow cytometry, will help illustrate these advances. PCR, the Nobel Prize winning technology, has revolutionized modern science. Here we will illustrate its use in cancer medicine. In chronic myelogenous leukemia (CML), a type of leukemia affecting approximately 5 000 people annually, claims approximate 2, 500 lives per year. More than 95% of these patients have a molecular signature called Philadelphia chromosome. This is caused by swapping segments of chromosome 9 with those of chromosome 22. The detection of this feature is not only important for CML diagnosis, but also critical in evaluation of response to treatment and detection of recurrence. Traditionally, chromosome changes are detected via a special stain and subsequent microscopic examination. As a result, there are limitations in terms of the sensitivity of this test. With the emergence of PCR, even a few cells left behind after therapy can be detected [24]. Similarly, other newer technologies, such as flow cytometry and fluorescence in situ hybridization (FISH), have significantly advanced the diagnosis of cancer [25, 26]. In flow cytometry studies, with the aid of special antibodies against a panel of cell surface markers and a special machine, doctors can now categorize cancers according to patterns of cell surface markers, making it far more reliable than simple morphology [27]. There are many other examples for applications of newer laboratory technologies in cancer management. We will dedicate the majority of the following section to expression array analysis.

11.2.2. Expression microarrays

There are great needs in the pursuit of better ways to predict the outcomes of disease, especially, in the field of cancer medicine. Recently developed microarray techniques present unique opportunities to investigate gene function on a genomic scale, and provide a versatile platform for utilizing genomic information to benefit human health [28]. In particular, gene-expression microarray technology is becoming an essential tool in exploring cancer development. In this subsection, we will review the basic principles of the microarray technology as well as its applications in cancer research and management. We will highlight the current advances, mostly using breast cancer as an example.

Microarrays are used to survey the expression of thousands of genes in a single experiment. Applied creatively, they can be used to test and generate new hypotheses. As the technology becomes more accessible, microarray analysis is finding applications in diverse areas of biology and medicine. Microarrays are a method for visualizing genes likely to be expressed in a particular tissue at a particular time under a particular set of conditions. The output of a microarray experiment is either called a gene-expression profile or a sample molecular profile.

The principles of microarray applications. A microarray, or "gene chip," is an orderly arrangement of oligonucleotide probes, some companies use 25-mer, some use 60-mer, attached to a solid support, measuring the expression level of a gene by determining the amount of messenger RNA that is present [29]. Microarray facilitates large-scale surveys of gene expression in which transcript levels can be determined for hundreds, thousands, or even tens of thousands of genes in a single experiment simultaneously. With the latest Affymetrix chip designs, the entire genome (less than 45 000 genes) can be measured on a single array. All microarray experiments rely on the core principle that transcript abundance can be deduced by measuring the amount of hybridization of labeled RNA to a complementary probe. The idea of a microarray is simply to lay down a field of thousands of these probes in, where each probe represents the complement of at least a part of a transcript that might be expressed in a tissue. Once the microarray is constructed, the target mRNA population is labeled, typically with a fluorescent dye, so that hybridization to the probe spot can be detected when scanned with a laser. The intensity of the signal produced by 1000 molecules of a particular labeled transcript could be twice as bright as the signal produced by 500 molecules and, similarly, that produced by 10 000 molecules half as bright as one produced by 20 000 molecules.

So a microarray is a massively parallel way to survey the expression of thousands of genes from different populations of cells. An illuminated microarray is shown in Figure 11.6. Trivially, if fluorescence is observed for a gene in one population but not another, the gene can be inferred to be on or off, respectively. With appropriate replication, normalization, and statistics, though, quantitative differences in abundance as small as 1.1-fold can readily be detected using Agilent chip when having huge sample size. The output of all microarray hybridizations is ultimately a series of numbers, which covers a range of three orders of magnitude,

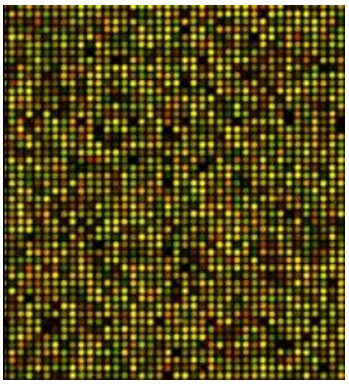


Figure 11.6. An illuminated Agilent microarray. A typical dimension of such an array is about 1 inch or less, the spot diameter is of the order of 250 μm to submicron for lithography-dependent processes, for some microarray types can be even smaller.

from perhaps less than one transcript per cell to many thousands of transcripts per cell. It is the comparison of gene-expression profiles that is usually of most interest. This is because the visualization is done at the level of transcript abundance, but just seeing a transcript does not guarantee that the protein is produced or functional. If, however, a difference in transcript abundance is observed between two or more conditions, it is natural to infer that the observed difference might point to an interesting biological phenomenon.

Many different design formats of microarrays exist, and the types of gene-expression assays include SAGE, complementary DNA (cDNA) arrays (e.g., Stanford University), fiber optic arrays (e.g., Illumina), short oligonucleotide array (e.g., Agilent inkjet), and long oligonucleotide arrays (e.g., Affymetrix). There are three main different methods for creating the microarray: (1) spotting long DNA fragments, (2) spotting prefabricated oligonucleotides, and (3) *in situ* (onchip) synthesis of oligonucleotides [30]. A presentation of type-1 is cDNA arrays, and type-2 is typically represented by Affymetrix arrays and Agilent arrays. The arrays require different targets, as cDNA is used for competitive hybridization. Spotted arrays make use of a complex biochemical-optical system to perform robotic spotting of cDNA probes that represent specific genes. Also, the design of probes is different, for instance, cDNA array uses the whole DNA sequence to prepare the probes, while oligonucleotide array uses a set of probes made by a segment of about 20 nucleotides. In comparison, the technology for spotting arrays is simpler than that for *in situ* fabrication, while oligonucleotide array is much more accurate and precise. Comparing arrays of prefabricated oligonucleotides and *in situ* synthesis of oligonucleotides, the latter has advantages over deposition of presynthesized oligonucleotides. Therefore, arrays of the third type are most widely applied. Two popular microarray platforms of this type include the Affymetrix array of 25-mer oligonucleotide probes, and the Agilent microarray consisting of 60-mer, *in situ* synthesized oligonucleotide probes. The Affymetrix arrays are based on a

photolithographic process, where the array is formatted by photolithographic synthesis of oligonucleotides [31]. The Agilent arrays are based on an *in situ* oligonucleotide synthesis method in which the inkjet printing process is modified to accommodate delivery of phosphoramidites to directed locations on a glass surface [32]. In both spotted and oligonucleotide arrays, the purified mRNA could be labeled by fluorescence or radioactivity, and then hybridized to the microarrays. Also, an imaging system is used to scan the hybridized arrays. After image processing, the normalization process is applied to each microarray.

Based on the input to the microarray, there are three applications of microarray technology: gene-expression profiling, genotyping, and DNA sequencing [29]. In the first case, mRNA extracted from a biological sample is applied to the microarray, and the gene-expression profile, the level of expressions of genes in that sample, is obtained. In the second case, a biological sample's DNA amplified by a PCR is hybridized to the microarray, and the genotype for genetic markers across the genome is determined [33]. In the third case, the DNA is applied to specific "sequencing" microarray, and thousands of base pairs of DNA are screened for polymorphisms in genes whose sequences are known [34].

The sophistication of microarray analysis blurs the distinction between hypothesis testing and data gathering very much. Hypothesis generation is just as important as testing, and very often expression profiling provides the necessary shift in a perspective that will fuel a new round of progress. In many gene-expression profiling experiments, the hypotheses being addressed are genome-wide integrative ones rather than single-gene reductionist queries. In general, without a hypothesis, only the most obvious features of a complex dataset will be seen, while clear formulation of the scientific question undoubtedly fuels better experimental design. And in some cases, the results of a microarray screen that was initially designed as an effort at cataloging expression differences are so unexpected that they immediately suggest novel conclusions and new areas of inquiry.

Investigators are interested not just in asking how large the magnitude of an expression difference is, but whether it is significant, given the other sources of variation in the experiment. Similarly, we might want to evaluate whether some subset of genes show similar expression profiles and so form natural clusters of functionally related genes. Or, we may combine expression studies with genotyping and surveys of regulatory sequences to investigate the mechanisms that are responsible for similar profiles of gene expression. Finally, all of the expression inferences must be integrated with everything else that is known about the genes, culled from text databases and proteomic experiments and from the investigator's own stores of biological insight. For instance, some genes are highly regulated at the transcriptional level and have low variability while others are highly variable from cell to cell but are highly invariable at the protein level.

The applications of microarrays in cancer study. In medical practice, it is very often that the strategies for treatment are tailored according to features that predict the disease outcome and response to a particular treatment strategy. It is therefore critical to use the most reliable predicting features available. Pathology of cancer

(histopathology) as well as clinical presentations such as tumor involvement of the lymph nodes and distant organs are some of the most useful ways used to predict cancer outcome. However, not all cancers presented at the same stage behave in the same way; better predicting tools are clearly needed. Microarrays offer a new way of categorizing cancers according to their molecular features (e.g., patterns of genes). The ultimate goal would be to use a small set of genes placed on a commercial available array to evaluate the molecular profile of cancer and predict prognosis and design treatment strategies accordingly. Moving cancer diagnosis away from visually based systems to such molecular-based systems is a major goal of the cancer research community.

The rationale behind this proposition is based on the belief that the overall behavior of a cancer is determined by the expressions of the genes that are modified. Microarray technology allows scientists to measure the activity of thousands of genes in a given cancer at one time. The active presence of specific genes can then be used by researchers to more specifically profile the tumor, and to predict the aggressiveness of the malignancy. The combination of gene-expression profiling and advanced bioinformatics is beginning to show promise in analyses of hitherto indistinguishable disease states. For patients who, on conventional clinical and histopathological criteria, have the same stage and grade of cancer, this refinement in tumor classification allows more accurate prediction of the course of disease. Using microarrays in human cancer study attracts increasing interest, the increasing availability and maturity of this technology has been leading to an explosion of cancer profiling studies, and it has the potential of impacting and revolutionizing the diagnosis, prognosis, and treatment of cancer patients.

The following section gives some specific examples of microarray's applications in cancer management to highlight the current advances and applications of microarray technology in cancer research, with a focus on breast cancer.

(a) *Molecular classification of tumors.* Tumor classification has been primarily based on morphological appearance, and has historically relied on specific biological insights. Serious limitations are associated with such an approach, since tumors with similar histopathological appearance can follow significantly different clinical courses and show different responses to therapy [35]. Particularly, the facts that breast tumors consist of many cell types and breast carcinoma (BC) cells themselves are morphologically and genetically diverse, make accurate classification of human breast tumors very difficult [36]. Recent genomic analysis approaches offer great promise for tumor classification based on gene expression monitoring by microarrays.

A hierarchical gene-clustering algorithm was used to identify gene-expression patterns in human mammary epithelial cells growing in culture and in primary breast tumors [36]. Clustering is defined as unsupervised classification of patterns into groups. It was found that systematic characterization of gene-expression patterns is useful to classify breast tumors into categories associated with cell types and response to specific physiological and pathological perturbations. The potential of using the microarray data for identifying previously uncharacterized tumor

subgroups has been shown in [35, 37]. A class discovery procedure, based on a technique called self-organizing maps (SOMs), automatically discovered the distinction between different leukemia groups, that is, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) [35]. SOM is an unsupervised neural networks technique. Alizadeh et al. [37] extended this class discover approach to identify two molecularly distinct categories of diffuse large B-cell lymphoma (DLBCL). Clustering analysis on microarray profiling has also been employed to distinguish the cancerous and noncancerous specimens [38], where a two-way clustering method based on the deterministic-annealing algorithm was developed to organize the data in a binary tree. cDNA array was used for classifying normal breast epithelial cells, invasive cells, and metastatic cells [39]. An analysis procedure was presented in [40] for classifying (predicting) human tumor samples based on microarray gene expressions. The proposed procedure involves dimension reduction using partial least squares (PLSs) and classification using logistic discrimination (LD) and quadratic discriminant analysis (QDA). Artificial neural networks (ANNs) were applied to build cDNA- and oligonucleotide-based tumor classifiers capable of deciphering the identity of most human cancers [41].

The above results show that genomic analysis based on gene-expression microarray data provides a systematic and unbiased approach to defining each individual property of a tumor and understanding its diagnosis and clinical behavior. However, it is worth mentioning that we are still far from a complete understanding of the diversity of many malignancies and there are potential limitations of the microarray approaches, since human tissues are complex and highly variable in their histology (e.g., the specimens may represent mixtures of cell types). For example, the two new DLBCL groups obtained in [37] were not heterogenic monomorphic (identical morphologically). Therefore, microarray technology is not meant to replace other classical technologies in cancer study. Instead, microarray data will be integrated with other data sources (e.g., clinical data, drug sensitivity data, and data from new techniques such as tissue array) for this purpose. With the vast amount of work still to be done, microarray technology will have an increasingly important role in cancer study.

(b) *Prediction of prognosis and tumor response to specific therapies.* Breast cancer is a heterogeneous disease encompassing different morphological types and clinical presentations. However, these specificities are currently difficult to incorporate into decision making with respect to therapy, and some simple prognostic factors, such as tumor size, lymph node involvement, grade, and hormonal receptor status, are commonly used. These currently used prognostic factors can only imperfectly predict therapeutic outcome. Microarray technology can, in principle, provide many types of information to help prediction of prognosis and tumor response to specific therapies in cancer treatment, and thus is expected to have a revolutionary effect on cancer clinical practice by leading to a personalized, safer, and more efficient approach to therapy.

Information generated from expression microarray could help reduce unnecessary therapy in patient with very good prognosis, sparing patients' exposure to

an ineffective and often toxic treatment, reducing the overall treatment cost, and planning treatments in accordance with the probability of success in patients. Examples include identifying tumors that are likely to be responsive to specific cancer treatments and identifying individuals who are likely to experience toxicity or drug metabolism effects (i.e., mutations in p450).

Gene-expression profiling can be used for predicting responders and nonresponders to chemotherapy [42]. Systemic chemotherapy substantially reduces the risk of death in breast cancer patients. So far, there are no efficient methods to distinguish between responders and nonresponders. In [42], with a list of discriminatory genes and their associated t values, a linear classifier based on the compound covariate predictor method was developed to predict response to docetaxel chemotherapy (e.g., responsive or resistant).

In addition to studying genes, it is now possible to look at the array of proteins that leave complex patterns in the blood and urine of cancer patients. Researchers have reported promising results for a potential test to detect ovarian cancer in a single drop of blood. In the initial experimental group, all 50 ovarian cancer patients in the test group were identified correctly, including 18 that were early stage and thus highly curable. The test also correctly identified 63 of 66 noncancerous samples [42, 43]. If these results hold up in a larger sample, it will be a great step forward in the early detection of ovarian cancer.

The above research suggests that gene- or gene-product expression arrays could be used to predict the effectiveness of treatment. However, further studies are needed to refine the molecular expression fingerprint by which to portrait each tumor and predict the response.

(c) *Drug development and identification of therapeutic targets.* The potentials of molecular profiling by microarrays goes far beyond diagnosis. It also provides researchers with clues on how to design therapies to target tumors at their most vulnerable points—the genes or proteins that are deranged in cancer. Biochemistry-based approach has traditionally served the purpose of drug discovery and development. Although the advent of molecular biology changed the process of drug discovery, some major barriers remain in place [39]. The recent microarray technology provides promise to surmount these barriers, and can be essential to help drug discovery and development.

Microarrays are powerful tools for investigating the precise mechanisms of action for a drug. For instance, microarrays can be used to screen for changes in gene expression following exposure of tumor cells to drugs. In [44], by combining the gene-expression data with molecular pharmacology data, the authors constructed the relationships between a set of 188 drugs based on the transcriptional activity pattern of the drugs against the NC160 cell lines. A yeast model system was used to validate drug targets, identify secondary drug targets [45], and to investigate the mechanism of drug action [46].

Expression microarray technology can be used to generate information rapidly for the identification and validation of novel therapeutic targets. For instance, in [47], the expression profiles of over 60 000 genes were measured for 15 infiltrating

ductal carcinoma (IDC) and 13 normal human breast tissues. Fold-change comparison between normal and IDC tissue samples identified 830 candidate genes. Further analysis utilizing principle components analysis and hierarchical clustering revealed tissue-specific candidate targets. This study provides a rational basis for the identification of therapeutic targets and diagnostic markers using microarray technology.

In summary, the results that have been achieved so far clearly demonstrate the feasibility of cancer diagnosis based on microarrays and suggest a general strategy for discovering new subtypes of cancer, independent of previous biological knowledge. It suggests that we may be able to use DNA microarrays in a variety of clinical settings for confirming diagnoses, clarifying unusual cases, and predicting patients' response to drugs. It may also provide an objective way to design therapies and predict clinical outcome for patients. In principle, gene activities that determine the biological behavior of a tumor are more likely to reflect its aggressiveness than general parameters such as tumor size and age of the patient. The clinical consequence is that treatments can be tailored according to the gene-activity patterns of the tumor.

No one doubts that this new technology will have a major impact on cancer management. A comprehensive molecular understanding of tumor formation may transform cancer from a death sentence into a chronic, but manageable disease.

Further analysis approaches are needed before the microarray technology can become a practical tool for identifying therapeutic targets and monitoring drug action. It is becoming increasingly necessary to integrate expression data with other types of data to gain new insights. For instance, microarray expression data can be used in conjunction with database of drug sensitivity to unravel the molecular basis of drug action.

11.2.3. Pharmacogenomics

Pharmacogenomics is the study of how an individual's genetic inheritance affects the body's response to drugs. The term comes from the words pharmacology and genomics and is thus the intersection of pharmacology and genetics. Pharmacogenomic analysis help examining the inherited variations in genes that dictate drug response and explore the ways these variations can be used to predict patients' response or lack thereof. It may also be possible to predict adverse effects along the same lines.

Understanding of human genetics and different variations between individuals is thought to be the key to creating personalized drugs with greater efficacy and safety. Right now, there are intense efforts to catalog as many of the genetic variations found within the human genome as possible. These variations, or single-nucleotide polymorphisms (SNPs), pronounced "snips," as they are commonly called, can be used as a diagnostic tool to predict a person's drug response [48]. For SNPs to be used in this way, a person's DNA must be examined (sequenced) for the presence of specific SNPs. The problem is that traditional gene-sequencing technology is very slow and expensive and has therefore impeded the

widespread use of SNPs as a diagnostic tool. DNA microarrays (or DNA chips) are an evolving technology that should make it possible for doctors to examine their patients for the presence of specific SNPs quickly and affordably. A single microarray can now be used to screen 100 000 SNPs found in a patient's genome in a matter of hours. As DNA microarray technology further matures, SNP screening in advance of drug administration to predict patients' response would be quite possible.

Many therapeutic agents in cancer treatment have limited efficacies. In addition, nearly all of them also have some degree of adverse effects or toxicities. These toxicities sometimes can be quite severe. However, the fatal consequences of not treating cancer make it impossible for doctors to withhold these therapies despite their imperfection. For example, paclitaxel is a relatively new chemotherapeutic drug with wide range of activities against a variety of cancers. Even so, response rates for many cancers to paclitaxel are less than 50%. In nonsmall cell lung cancer, for example, a paclitaxel-containing regimen only produce 30–40% responses as initial therapy [49]. This means that it is less than efficacious in 60–70% of the patients. Additionally, potentially fatal side effects such as anaphylaxis (generalized allergic reaction) and nonfatal adverse effects such as neuropathy (nerve damage) are hard to predict on an individual basis. Pharmacogenomics holds promise to solve problems posted by drugs such as paclitaxel and many others. It could be of great value not only in predicting how cancer would respond to a therapy, but also in predicting how the host will respond to treatment. The ideal therapy would boast great efficacy in treating cancer without any significant adverse effects to the host.

11.3. New paradigm in cancer therapeutics

The ultimate goal in cancer research is to maximize our understanding of the disease and to improve the quality and outcome of cancer care. Although surgery, radiotherapy, and chemotherapy are the current major pillars to cancer treatment, many recent advances focus on new, molecularly targeted therapeutic modalities. Unlike older generation of cancer therapies, which are less differentiating of cancer versus normal cells, newer generations of cancer treatments boast more specificity, efficacy, and less toxicity. The following discussion will concentrate on some of these new treatment strategies.

11.3.1. New targets for cancer therapy

Better understanding of molecular events inside cancer cells has led to improved strategies in designing cancer treatments. For the past few decades, cancer therapy has been generally focused on some of the general biologic properties of cancer cells, such as the rapid rate of replication, as ways of therapeutic approach. Molecular targeting focuses on what is distinct inside the cancer cells and offers much better efficacy and specificity. Below are some highlights of current advances to illustrate this.

(1) *Telomerase as a target for cancer therapy.* Telomerase is a special enzyme that is expressed in most cancer and immortal cells [50]. It has a unique capability of stabilizing and extending the end portion of the DNA-telomere, and hence maintains the genomic integrity of cells [51]. This contributes partially to the unlimited division potential of cancer cells. The detection of telomerase activity is of a diagnostic and prognostic value in many cancers. Typically cancer cells but not normal somatic cells express it, drugs developed against telomerase may prove to be highly selective and free from toxic side effects. There has been considerable interest in elucidating mechanisms that regulate the capacity for cell division in normal as well as cancer cells, and substantial attention has focused specifically on the role of telomeres as the "mitotic clock" that mediates replicative capacity. A number of approaches have been taken to inhibit growth of malignant cells by targeting their expression of telomerase.

(a) One straightforward strategy has been the inhibition of telomerase activity in malignant cells [52, 53]. Inhibition of telomerase activity is expected to result in progressive telomere shortening during cell division and would eventually lead to cessation of replication and cell death. The downside for this kind of strategies is that it might not be effective early on, and that tumor cells would continue to grow until telomere shortening reached a critical level before growth arrest and apoptosis (see below) occur.
(b) An alternative strategy, one not aimed at inhibition of telomerase enzymatic activity, but rather at promoting immune recognition and destruction of cells that express telomerase, would result in rapid immune elimination of telomerase-expressing tumor cells without the lag time involved in strategies that inhibit telomerase function and depend on gradual telomere shortening to inhibit tumor growth [54].
(c) Another approach to the targeting of telomeres in cancer cells employs the strategy of expressing a mutant telomerase template RNA to inhibit tumor cell proliferation [55]. It does not involve either telomere shortening or inhibition of telomerase activity. It is rapid in onset and can exert immediate impact on tumor cells. Although the exact mechanism for this is still not clear, several theories have been proposed. It is speculated that incorporation of mutant sequences into telomeric DNA interferes with telomere function, possibly through altered DNA-protein or DNA-DNA interactions, and triggers inhibition of proliferation and induction of apoptosis.

In general, although targeting the telomerase enzyme as part of cancer treatment is still at the preclinical stage, it offers great promise for the near future. New insights into the role of telomere function in cell survival and cell cycle regulation will further help in designing effective therapy in targeting telomerase.

(2) *Interfering the survival pathways.* Normal cells go through their life cycles of division, growth, maturation, and death. This life cycle is mainly dictated by the

cells' genetic codes as well as their environment. During this process, cells may undergo apoptosis (self-destruction) under certain circumstances [56]. Triggers for apoptosis are numerous. For example, drugs and radiation can lead to cell damage and thus turn on genes that lead to apoptosis. Apoptosis sometimes can be viewed as a self-protection strategy employed by the host to eliminate abnormal or damaged cells. Abnormalities of genes regulating apoptosis can lead to uncontrolled cell growth and increase the incidence of cancer. For example, amplification of genes that are antiapoptotic, such as bcl-2, can lead to decrease in apoptosis and therefore promote tumor growth [57]. On the other hand, mutation/deletion of genes that are proapoptotic, such as p53, can also lead to decrease of apoptosis and therefore promote cancer formation [58]. In fact, both overexpression of bcl-2 or mutation of p53 are quite common in human cancer. Because of this, it is rational to think that agents aimed at modifying these genetic abnormalities can be effective in cancer therapy. Currently, many studies are underway to explore the possibility of manipulating apoptotic pathways in cancer therapy. For example, a drug named Genasense^{TM} or oblimersen sodium is currently under phase III study [59]. It is a bcl-2 antisense oligonucleotide. Proteins are made according to transcribed codes in DNA. Protein are synthesized from mRNA created during transcription from DNA. Antisense oligonucleotides work by binding to the mRNA, therefore blocking its translation into protein, in this case bcl-2 protein.

(3) *Targeting tumor angiogenesis.* In the early 1970s, Dr. Judah Folkman pioneered the research to help understand the mechanisms of angiogenesis (new blood vessel formation) and its potential application in clinical medicine. But it was not until the 1990s that this field finally took off. Many new agents, either inhibitors of proangiogenesis molecules or antiangiogenesis agents, are in various phases of research and clinical trials. The principle behind antiangiogenesis therapy is that in order for a tumor to grow, it must have sufficient blood supply. Many tumors are able to induce new blood vessel formation in order to obtain enough blood to support their rapid growth. By inhibiting the formation of new blood vessels, the cancer cells starve. Currently, there are a few such drugs that are either already approved or are on the verge of approval for anticancer therapy. For example, thalidomide, once a notorious drug implicated in causing birth defects when taken by pregnant women and hence removed from the market, was revived because it was found that it had antiangiogenesis properties. In fact, the reason why it caused fetus deformities was probably due to its antiangiogenesis property. Currently, thalidomide is found to be effective in treating multiple myeloma and thus was approved for this use by the FDA [60]. Of course, its use needs to be closely monitored because of this teratogenic effect. Newer generations of drugs that work in a way similar to thalidomide (so-called IMiDs, or immunomodulating drugs) are being developed. One of them known as Revlimid is currently in late-phase clinical trial and looks very promising. Another example for drugs with antiangiogenesis property is bevacizumab (Avastin^{TM}). It is a monoclonal antibody against vascular endothelial growth factor (VEGF), a very important molecule for new

blood vessel formation. In a recent clinical trial, bevacizumab in combination with chemotherapy has been shown to be superior to chemotherapy alone in patients with metastatic colon cancer [61]. Bevacizumab has recently been approved by the FDA for treatment of metastatic colon cancer in combination with chemotherapy. Clinical trials for the use of bevacizumab in other types of cancers are on going.

(4) *Targeting growth factors and their receptors.* Growth factors to cells are like fuels to automobiles. They are essential for cell survival and growth. One instinctive feature of cancer cells is uncontrolled growth and loss of contact inhibition. Many types of cancer cells have excessive or aberrant growth factor receptors on their surface which lead to this unregulated growth. Targeting growth factors and their receptors thus represent an important notion. Some growth factor inhibitors are starting to be used in clinical practice. For example, Cetuximab (also known as Erbitux or C225) is a monoclonal antibody against epidermal growth factor receptor (EGFR) [62]. It binds EGFRs and inhibits their activation. Another example, Gefitinib (also known as Iressa) is a small molecule also inhibit EGFR activation (refer to details later in the chapter). It is sufficient to say that inhibitors for growth factors and their receptors are major targets for future development of new cancer therapies.

(5) *Targeting intracellular signal pathways.* Signal transduction pathways are essential channels that cells rely on to communicate with the outside environment. Once activated, they relay environmental stimuli into the cells via signaling pathways and allow cells to react properly to different stimuli. They are highly regulated so that cells are constantly instructed to act according to appropriate guidelines. In contrast to normal cells, signal transduction in cancer cells are often disregulated. For example, a component of the signal transduction pathway can be constitutively activated and this leads to uncontrolled and unregulated cell growth. This is the case with the oncogene ras in many types of cancers. Ras is a guanine nucleotide-binding protein. When in its active form, it binds guanine triphosphate (GTP) and leads to hydrolysis of GTP to guanine diphosphate (GDP) and thus becomes inactive. When the *ras* gene is mutated at certain positions, it can no longer induce the GTP to GDP conversion. It stuck with GTP and remains constitutively active. This leads to uncontrolled activation of this signal transduction pathway and subsequent cancer growth. Research aimed at targeting numerous different signal transduction pathway components. Farnesyl transferase inhibitor for the ras oncogene is one of the examples of drugs targeting signal transduction and is currently under investigation.

(6) *Interfering intracellular protein synthesis, transport, and breakdown.* The integrity of cells is maintained by many intracellular proteins. This protein network is tightly regulated through controlled synthesis and breakdown. Excessive proteins or constitutive activation or deactivation of certain proteins can contribute to the development of cancer. New generations of cancer therapy also target some of the key pathways in protein synthesis and degradation. It is well known that NF-κb is one of the transcription factors important for normal and cancerous cell

growth and survival. It normally exists in the cytoplasm of cells and is bound to I-κb, an inhibitor of NF-κb, thus remaining inactive. Degradation of I-κb can free NF-κb from I-κb and lead to cell growth. Bortezomib (also known as Valcade or PS341) represents a new class of therapeutic called the proteosome inhibitor. It acts essentially by blocking proteosome from doing its job, protein degradation. It is thought that at least part of the anticancer mechanism for Valcade is due to the inhibition of I-κb degradation and thus indirectly causes more inhibition of NF-κb. In the pivotal trial that led to its approval, Valcade has been shown to be remarkably effective even on patients with refractory multiple myeloma who have gone through regimens [63, 64]. It is reasonable to believe that more anticancer therapeutic agents of this class will be developed in the future.

11.3.2. New agents and approaches

(1) *Monoclonal antibody.* Substances foreign to the body, such as disease-causing bacteria, viruses and other infectious agents, known as antigens, are recognized by the body's immune system as invaders. One kind of natural defense against these infectious agents are antibodies, proteins that seek out the antigens and recruit factors that eventually destroy the foreign substance. Antibodies have two very useful characteristics. First, they can be very specific; that is, each antibody typically binds to and attacks one particular antigen, either directly or indirectly. Second, some antibodies continue to confer resistance against specific disease for life; classic examples are the antibodies to the childhood diseases chickenpox and measles. The second characteristic of antibodies makes it possible to develop vaccines. A vaccine is a preparation of killed or weakened bacteria or virus that, when introduced into the body, stimulates the production of antibodies against the antigens it contains.

Although each kind of antibody reacts to one particular kind of antigen in many of the circumstances, many antibodies can be generated by a single disease entity. Not all are useful as therapy and it is difficult to reproduce a pure and highly specific antibody in large quantity. The conventional method of producing an antibody was to inject a laboratory animal with an antigen and after antibodies had formed, collect those antibodies from the blood serum (antibody-containing blood serum is called antiserum). There are two problems with this method: it yields antiserum that contains contaminants, and it provides a very small amount of usable antibody. Monoclonal antibody technology uses a different approach to produce antibodies. By fusing antibody-producing B-cell with a kind of immortalized cell, a hybridoma is produced [65]. Many clones of these hybridomas are isolated and tested for antibody production. Those clones that produce large quantities of pure antibodies are then expanded in cell culture dishes.

Monoclonal antibodies have been produced for different clinical uses and many have been proven to be more effective than conventional drugs in fighting disease. Monoclonal antibodies attack the target molecule more specifically, thus greatly diminishing the side effects associated with many of the conventional drugs. Some examples of monoclonal antibodies include Rituximab, an anti-CD20

antibody that is approved for treatment of B-cell non-Hodgkin's lymphoma (NHL) and also used off label for other clinical entities; Trastuzumab, also known as herceptin, is another kind of monoclonal antibody against Her2/*neu*. It is used to treat a sub-population of breast cancer. The specificity of monoclonal antibodies makes them valuable in the laboratory as well as in clinical medicine. Not only can antibodies be used therapeutically to fight against disease, they can also help to diagnose a wide variety of illness, and can detect the presence of drugs, viral and bacterial products, and other unusual or abnormal substances in the blood.

(2) *Designed small molecules.* Unlike antibody-based therapies, small molecules offer some advantages, at least partially by the virtue of their size. The smaller size allows easier synthesis, better delivery, and potentially better efficacies in tumor penetration. Small molecules can also be specifically designed to function like a key to a lock, exerting a function either positively or negatively by turning on or off a molecular pathway. Below are examples of small-molecule-based cancer therapies.

(i) **Imatinib.** In 2001, the FDA approved the oral medication imatinib (Gleevec^{TM}) for CML, a type of leukemia that accounts for 7–20% of all adult leukemias. Imatinib is one of the first small molecule drugs approved for clinical use in cancer treatment. It is designed uniquely to interact with a class of receptor tyrosine kinases (RTKs) including PDGFR (platelet-derived growth factor receptor), c-kit (CD-117), and bcr-abl tyrosine kinases, a family of important enzymes crucial for cell signaling. Receptors that possess tyrosine kinase activities are called RTKs. In CML, over 95% of the cancer cells express bcr-abl tyrosine kinase. This tyrosine kinase is a hybrid protein resulting from chromosomal translocation. This unique marker for CML cells allows for the specific targeting of cancer cells while sparing normal cells. It works at the molecular level by blocking the phosphorylation and subsequent activation of bcr-abl tyrosine kinase, effectively shutting down what drives the abnormal growth of CML cells. Response rate for treatment of CML are up to 80–90%, a great improvement from traditional regimens [66]. Equally amazing is how fast imatinib works. Imaging studies taken a few days after initiation of therapy have shown complete silencing of tumor activity [67]. So far, the side effects of the drug are well tolerated. Additional investigations have shown it is also effective against a previously untreatable form of gastrointestinal cancer called gastrointestinal stromal tumor (GIST) because it inhibits another RTK, expressed by a majority of the GIST cells [68]. Currently, imatinib is being tested in a variety of tumors that overexpress PDGFR. Its use in cancer may be further expanded depending on the results of future clinical trials.

(ii) **Gefitinib.** Gefitinib (Iressa) is another type of small molecular that works by blocking the RTK that is associated with tumor growth. It works by inhibiting a different type of RTK called EGFR. Gefitinib has recently been approved for third-line therapy for nonsmall cell lung cancer [69, 70]. It has also been widely tested for treatment of many other types of cancer. It is important to point out that Gefitinib only produced a response rate of 10–20% as second- or third-line

therapy for lung cancer, and so far, the level of EGFR expression did not correspond with the degree of response. Recently, mutations in EGFR were identified that appear to correlate responses to Gefitinib [71]. This again suggests the importance of newer strategies in determining drug responses and selecting patients for treatment.

(3) *Radioimmunotherapy.* Radiation is one of the most important treatment modalities widely used in almost every type of cancer. It can be given externally by focusing a "beam" of radiation on cancer or cancer involved organ from outside the body. This is commonly referred as external beam radiation. Alternatively, radiation can also be given by implanting radioactive seeds inside the cancer-containing organ (such as prostate gland). This is commonly referred as brachytherapy. Radioimmunotherapy, on the other hand, is the delivery of radioactivity through immunotargeting. It essentially delivers radiation via coupling of radioactive isotope with Monoclonal antibodies. Monoclonal antibodies bind target cells more specifically, therefore they can guide the radioisotope to specific locations on a cellular level. Radioimmunotherapy in cancer treatment targets tumor cells throughout the body, but it remains considerably toxic.

Radiation therapies given by external beam or by brachytherapy requires that the tumors are geographically limited to a region that can be efficiently targeted. It cannot be used if the tumor is widely disseminated. Too large of a radiation field also significantly increases toxicity and limits the tolerance of the host. Radioimmunotherapy, on the other hand, uses antibodies to seek for tumor cells in various locations throughout the body. It is therefore more useful for tumors that are widely distributed, such as leukemia or lymphoma.

Radioimmunotherapeutic agents are like double-edged swords. Monoclonal antibodies can affect the cancer cells by themselves even without linkage to radioisotope. They can inhibit tumor cell growth and induce apoptosis. With the addition of radioisotope, it adds more "firepower" by delivery of radiation to the target cell at the same time. This dual-action effect renders this therapy more potent and effective in cancer treatment.

Currently, there are two radioimmunologic agents approved by the FDA for radioimmunotherapy for NHL: Ibritumomab Tiuxetan (Zevalin) and Tusitomomab (Bexxar). Both utilize the power of different radiation particles called isotopes. The radioactive particles in Zevalin emit beta radiation, which travels over a relatively short distance. The radioactive particles in Bexxar give off beta and gamma radiation. The gamma radiation travels a longer distance.

Both Zevalin and Bexxar are dual-action drugs that pair the tumor-targeting and killing ability of monoclonal antibodies (anti-CD20) and therapeutic radiation (Yttrium-90 and Iodine-131, respectively). Combined, these agents are much more potent than either anti-CD20 monoclonal antibodies or radiotherapy alone for NHL [72, 73, 74]. Although monoclonal antibodies confer specificity for tumor cell targeting, it is unavoidably that they also affect some normal cells due to the fact that normal cells can express CD-20 and the radioisotope affects "innocent bystanders."

In addition to the increasing use of radioimmunotherapy in cancer treatments, radioimmunologic agents can also be used for cancer diagnosis.

(4) *New invasive therapeutic modalities.* Although current research on cancer diagnosis and treatment weigh heavily toward less traumatic and invasive approaches, this does not mean that the invasive methods are of less value. Actually, newer and better tolerated invasive therapies are becoming increasingly available as technologies advance. More and more interventions are done by an interventional radiologist under imaging guidance. These new strategies are also being applied to cancer diagnosis and treatment. Radiofrequency ablation (RFA) is a relatively new therapy for cancer treatment. It is a treatment strategy based on the application of heat energy [75]. RFA causes the cellular destruction of tissues by heating them. Using various imaging methods (ultrasound, CT scan, or MRI) as guidance, the procedure involves placing a needle through the skin and into the tumor under local anesthesia. Once positioned in the targeted treatment area, an umbrella-like array of electrodes is deployed and radio waves are generated through agitation caused by alternating electrical current (radiofrequency energy) moving through the tissue. The heat generated in this fashion results in local cell coagulation and demise. Normal tissues surrounding the lesion are usually spared due to the limited range of heating. Destroyed cells are reabsorbed by the body over time.

In RFA, since the patient's body is only penetrated with a special needle, it is minimally invasive and the procedure is performed under local anesthesia and conscious sedation, therefore most patients will be able to return home the day of the procedure. There are other advantages of RFA besides better tolerance. For example, patients with oligometastasis (a few lesions rather than many lesions) in both lobes of the liver could not undergo resection, RFA would then be a reasonable option. Another example, a patient with a single, solitary kidney tumor who is too ill to tolerate a surgical procedure, could choose to have RFA instead. Other applications of RFA include symptom management such as relieving pains caused by tumor growth. In some circumstances, chemotherapy can be given in conjunction with RFA to maximize the efficacy and increase the range of tumor cell killing.

Despite the advantages mentioned above, there are still many limitations to RFA treatment. First, the tumor has to be accessible by needle. This means that tumors located deep inside the human body may not be amendable to this treatment. Second, it will not be able to kill tumors more than 5 cm in diameter due to the limitation of heat travel and dissipation. Third, multiple lesions, typically more than 3 lesions, cannot be effectively treated by RFA. Fourth, potential risk including bowel perforation, or damage to other important structures, makes it important to select the patient carefully. Fifth, improvement of survival is possible as a result of RFA but has not been definitively proven.

RFA is not intended to replace surgery and/or chemotherapy. Instead, it is designed to work in conjunction with these modalities. Its current application still remains as adjunct to other treatment strategies.

In summary, RFA is a minimally invasive method that can be used to treat multiple types of cancers. It is ideal for treating oligotumor lesions (usually not more than 3), and relieving the symptoms caused by tumor growth. RFA has been proven to be a very valuable strategy in the toolkit of cancer therapy and give patients another therapeutic option.

11.4. Nanotechnology in cancer study

Nanotechnology has been an emergent multidisciplinary research topic with applications in many areas, such as nano-electro-mechanical systems (NEMSs) [76], nanocomputers, and nanoscale medicine [77]. In this subsection, we will elaborate how nanotechnology can be applied in cancer study.

A nanometer is one billionth of a meter—1/80 000, which is the width of a human hair, or about the combined diameter of ten hydrogen atoms. Nanotechnology is about how to create useful materials, devices, or systems through the manipulation of miniscule matter. It will also greatly impact biotechnology. For instance, two grams of DNA can hold as much information as the whole Library of Congress. A carbon-nanotube-based nanoprobe on the 1 nm scale enables us to process the information encoded in DNA. The interdisciplinary nature of nanotechnology involves scientists from many different disciplines such as physics, chemistry, engineering, biology, and even sociology.

Currently, nanotechnology is a nascent research area. Nanoscale fabrication is far from mature. Most of the research focuses on basic material preparation, such as making different carbon nanotubes and molecular devices [78, 79]. There are two basic approaches to creating nanostructures. The first method is "top-down" approach, which involves molding or etching materials into small components. This approach has traditionally been used in making computer and electronic devices. The second approach is "bottom-up" approach. The bottom-up approach involves assembling structures atom-by-atom or molecule-by-molecule. Protein synthesis by ribosome is a good example of a bottom-up approach. Different self-assembly techniques are currently being investigated in order to reach a better understanding of how to integrate nanoscale building blocks [80].

Material behaves differently on the nanometer scale than it does on larger scale. Physical properties governing larger systems do not necessarily apply to nanoscale systems. Electron propagates as a wave instead of as a single particle, thus quantum effects have to be taken into design considerations [81]. Because nanomaterial, such as carbon nanotube, has large surface area relative to its volume, phenomena like friction become more important at the nanometer level than in larger systems. These physical property differences will affect nanosystem design for biomedical applications.

Nanostructures are so tiny that they may be easily washed away before they can take effect in cancer diagnosis or imaging. On the other hand, larger nanoparticles may accumulate in vital organs and can potentially cause the discomfort of patients, sometimes even resulting in toxicity. Scientists are trying to find the solutions of these problems. They study how nanostructures will behave in human body and attempt to create devices that can easily adapt to *in vivo* circumstance.

11.4.1. Applying nanotechnology in handling tumor cells

Most animal cells are 10 000 to 20 000 nm in diameter. This means that nanoscale devices (having at least one dimension less than 100 nm) can enter cells and organelles inside cells. These devices can interact with DNAs and proteins because the DNA double helix is 0.5 nm in diameter and the distance between each base pair is about 1.5–2 nm. Nanotechnology tools are very sensitive and are able to detect disease by using a very small amount of cells or tissues. These tools may even be able to enter cells and monitor cell activities. This provides a vital solution for us to study cancer genomics and proteomics, enabling us to understand cancer mechanisms more precisely.

With respect to tumor diagnosis, prognosis and treatment, miniaturization becomes necessary because nanoscale tools for many different tests need to be mounted on a same small device. This makes it possible to run multiple diagnostic tests simultaneously.

Cancer detection at its early stage is critical in improving cancer treatment. Cancer detection and diagnosis to date depend on the detection of abnormalities at tissue or organ level by physical examinations or imaging studies. Patients' mortality is often due to late finding and ineffective curative therapy. Ideally, scientists would much prefer to make it possible to detect cancer when the earliest molecular changes are present, long before it is large enough to be detected by physical examination or imaging technology. To do this is not an easy task, which needs a new set of tools at the very least. Nanotechnology is very promising in meeting these challenges based on the following reasons.

(i) To successfully detect cancer at its earliest stages, scientists must be able to detect molecular changes even when they occur only in a small percentage of cells. This means the nanotechnology tools must be extremely sensitive. The potential for nanostructures to enter and analyze single cells suggests they could meet this challenge.

(ii) Many nanotechnology tools will make it possible for clinicians to run tests without physically altering cells or tissues taken from a patient. This is important because the amount of samples that clinicians use to screen for cancer are often limited. Scientists would like to perform tests without altering cells, so they can be used again if further tests are needed. Reduction in the size of tools means that many tests can be run on a single small device. This will make screening fast and cost-efficient.

The following are some specific nanotechnology tools developed for early tumor detection.

(1) *Cantilever.* The cantilever is one of the nanotechnology tools with potential to aid cancer diagnosis. Nano-scale cantilevers are based on the simple mechanic concept: disparate materials bond together causing a mechanical reaction, much as bimetallic strips used in peltier coolers. Tiny bars anchored at one end of cantilever can be engineered to bind to molecules associated with cancer, or refer to [3] Protiveris' microcantilever technology (see www.protiveris.com). They may bind

to altered DNA sequences or proteins that are present in certain types of cancer. When the cancer-associated molecules bind to cantilevers, changes in surface tension cause cantilevers to bend. This nanoscale tool is very sensitive and is able to distinguish microreflections. Based on the observation, scientists can determine whether cancer molecules are present even when the altered molecules are present in very low concentrations. This tool is useful in detecting early molecular mutations during the course of cancer development.

(2) *Nanopores.* Nanopore device is another type of nanotechnology tools for reading genetic code on single strands of DNA. The idea was originally proposed by D. Branton group at Harvard University. Nanopores are tiny holes that connect two liquid compartments that are positively charged at one side and negatively charged at the other. The electrical field helps to ionize the single-strand DNA and makes it easy for DNA to pass through the nanopore. We can imagine DNA nucleotides (A, T, C, and G) look like beads on a string. As a DNA passes through a nanopore one nucleotide at a time, it causes a subtle electrical current difference (at the scale of $10^{-12}A$). Based on the different readings of four different base pairs that make up genetic code, scientists can decipher encoded DNA information, including mutations in DNA known to be associated with certain type of cancers. One of the major advantages to use nanopore design is that it makes long DNA sequencing possible. The conventional for DNA sequencing in biology is slow and only can handle short DNA sequence. The efficient nanopore design will significantly reduce lab work and make real-time *in vivo* DNA sequencing possible.

(3) C_{60} *bulky balls and nanotubes.* Since the discovery a decade ago [78, 82], both C_{60} and carbon nanotube have emerged as promising candidates for nanomedicine. C_{60} has been applied for drug delivery by encapsulating drug within C_{60} structure. By attaching different chemical compounds to the C_{60}, researchers are able to use it for cancer and AIDS treatments [83]. Nanotubes—carbon rods about half the diameter of a DNA molecule—also help identify DNA changes associated with cancers. Nanotubes are even smaller in diameter than nanopores. In addition to detecting the presence of altered genes, these materials may help researchers to pinpoint the exact location of these changes. For instance, mutated regions associated with a cancer are first tagged with bulky C_{60} molecules. These bulky molecules identify regions where mutations are present. These techniques will be important in predicting diseases. Once a target region is located, we can use a nanotube assembled on the needle tip of a record player to trace DNA's physical shape and its sequence. We can translate collected information into a topographical map, and compare the DNA reading with a database for cancer diagnosis and treatment at the genome level. Both C_{60} bulky balls and nanotubes are biologically inert materials that should not cause any side effects during diagnosis and treatment.

Besides biological applications, carbon nanotube is also very useful in nanoelectronic circuit design because of their unique electronic properties [84] and fine 1D structures. They exhibit either metallic or semiconducting behavior depending

on the diameter and helicity of tubes [85, 86]. It can also conduct current ballistically without associated heat dissipation [87]. They are very strong and insensitive to a wide range of processing temperatures and treatments.

(4) *Quantum dots.* Researchers would like to detect the early signs or precursors of a cancer in cells without removing them from body. One nanoscale tool that will allow scientists to detect critical DNA changes in vivo is via quantum dots. Quantum dots, tiny crystals that are much smaller than conventional fluorescent dyes, glow when they are stimulated by UV light [3]. Interestingly, the wavelength or the color of the emitting light depends on quantum dot size. Latex beads filled with these crystals can be designed to bind to specific DNA sequences. When crystals are stimulated by light, colors they emit act as dyes that light up the DNA sequences of interest. By mixing different-sized quantum dots within a single bead, scientists can create probes that release a distinct spectrum of various colors with different light intensities, serving as a sort of spectral bar code. To detect cancer, quantum dots can be designed to bind cancer DNA sequences. Based on the unique bar codes or labels, we can make critical cancer-associated DNA sequences visible. The vast number of possible combinations of quantum dots also means that many unique labels can be created simultaneously [88]. It is important in cancer diagnosis because it allows us to detect and interpret results from many different mutations within a cell.

(5) *Nanoparticles and nanoshells.* A number of nanoparticles have been developed to facilitate drug delivery. Researchers have developed an innovative way to encapsulate drug in gold nanoshells. These nanoshells are miniscule beads coated with gold. By manipulating the thickness of different layers making up nanoshells, a gold nanoshell can absorb specific wavelengths of light. The most useful nanoshells are those that absorb near-infrared light, which can easily penetrate several centimeters of human tissue. Light absorption by nanoshells creates an intense heat that is lethal to cancer cells. In laboratory cultures, heat generated by light-absorbing nanoshells has successfully killed tumor cells while leaving neighboring cells intact. To make nanoshells target at specific cancer, nanoshells can be linked to cancer-specific antibodies. Such a design can be envisioned as a "biological bomb" because these nanoshells can seek out their cancerous targets and bind with high affinity. When we apply near-infrared light, we can kill the tumor cells.

With state-of-the-art nanoscale fabrication techniques, some useful nanoparticles are developed for pharmaceutical, medical, chemical, and engineering applications. Cost-effective, reproducible, and scalable processes to engineer cell- or tissue-targeted nanoparticles are sought to deliver potent drugs as new therapies in the pharmaceutical field. A natural and spontaneous method to engineer nanoparticles has been developed through the use of microemulsions in which dispersed phase droplets serve as "nanotemplates" to directly form stable nanoparticles.

The other well-known means is "organic dendrimer" [3]. The useful feature of dendrimers is their branching shape that gives them vast amounts of surface area.

Therefore, it enhances the chance of delivery therapeutic agents or other biologically active molecules to desired locations. Because of the tiny branch geometry, it can also reach spots where a conventional drug cannot. A single dendrimer can carry a certain molecule that recognizes specific cancer cells. A therapeutic agent thus can kill these cells, and then another molecule can be applied to detect cell death. Researchers hope to manipulate dendrimers that can release their contents only when a certain type of cancer is present. Dendrimers may also feedback information about targeted tumor cells' apoptosis after drug released.

11.4.2. Case study: current application of nanotechnology in cancer management

The determination of a patient's cancer stage is of paramount importance in cancer therapy. It is important to determine whether a cancer has spread to a patient's lymph nodes or not in order to determine the diseases stage, and to design the best treatment to increase survival chance. Let us consider prostate cancer. If the cancer has spread to lymph nodes or bones, and the rogen-deprivation therapy is often needed for cancer therapy. Men whose prostate cancer is still confine within their prostate gland can select a range of loco-regional therapies, including radical prostatectomy by having the prostate removed along with the seminal vesicles and ductus deferens, radiotherapy (external beam or seeds), or watchful waiting (closely monitoring but deferring treatment until necessary).

MRI provides images with excellent anatomical detail and soft-tissue contrast. It is, however, relatively insensitive to detect lymph-node metastases. To improve the MRI results, different contrast agents and acquisition techniques can be applied. For instance, contrast agents, such as gadolinium, are routinely used to increase the accuracy of cancer staging by the MRI. Nanotechnology is very promising in this field, in particular, the use of lymphotropic superparamagnetic nanoparticles. These nanoparticles have a monocrystalline, inverse spinel, superparamagnetic iron oxide core, and contain a dense packing of dextrans to prolong their time in circulation. They are also avidly taken up by lymph nodes in animals and humans. Nanoparticles are slowly extravasated from the vascular into the interstitial space, from which they are transported to lymph nodes by lymphatic vessels. Within lymph nodes, lymphotropic superparamagnetic nanoparticles are internalized by macrophages, and these intracellular iron-containing particles cause changes in magnetic properties that is detectable by the MRI.

Accurate detection of lymph-node metastases in prostate cancer is an essential component of initial nanomedicine study. It can also be applied to other cancers. The means to identify men with clinically occult lymph-node metastases is greatly needed because of the adverse prognostic implications it confers. Highly lymphotropic superparamagnetic nanoparticles, which gain access to lymph nodes by means of interstitial-lymphatic fluid transport, have been used in conjunction with high-resolution MRI to reveal small nodal metastases [89]. It is demonstrated that high-resolution MRI injection with magnetic nanoparticles allows the detection of small and otherwise unsuspected lymph nodes involved in cancer progression.

Nanotechnology has promising potential for us to develop ways in eradicating cancer cells without harming healthy surrounding normal cells. Scientists hope to use nanotechnology to create therapeutic agents that target at specific cells while reduce toxic side effects in a controllable manner. The ultimate goal is to create an integrated solution that is able to both detect cancer and deliver treatment only to affected cells. In an ideal situation, nanoparticles that will circulate through the body could detect cancer-associated molecular changes, release therapeutic agent, and then monitor the effectiveness of intervention.

11.5. Conclusion and future directions

Medicine is evolving from what was essentially an observational science a century ago to what is largely a molecular science. The pace of this evolution is still accelerating. With this fascinating progress, a single disciplinary approach to understand cancer is no longer suitable. Collaboration between multiple areas of expertise such as scientists, engineers and physicians has become a hallmark of this new era. Not only is it important to have a team consisting of experts from various fields, it is also vital for each of these members in the team to understand their counterparts to maximize collaboration.

In this chapter, we reviewed the current understanding of cancer from a molecular point of view. Cancer genomics and proteomics form the backbone of our current knowledge. Microarray technology has fundamentally changed our approach to cancer in the past few years. It is safe to say that it will not be long before we will need to reclassify many types of malignancies according to data from gene profiling. More highly specific therapies against cancer would also be available as a result of our improved knowledge. As we focus our efforts in the microstructure of cells and their DNA, protein makeup as well as function, better tools for cancer detection, assessment and therapeutic delivery are important. Nanoparticles boast the advantages of being both extremely tiny and being biologically inert. The potential of having nanoparticles penetrating into the microstructures inside cells and performing multiple tasks will undoubtedly bring us to a new horizon.

Our multidisciplinary research focuses on two aspects: (1) a stochastic framework for modeling genomic/proteomic network structure and dynamics in p53-induced apoptosis; (2) carbon nanotubes as a remotely activated cytotoxic agent in tumor-suppression therapy.

11.5.1. Stochastic framework for modeling genomics/proteomics in p53-induced apoptosis network

The goal is to develop a holistic mathematical framework for understanding, evaluating and validating the regulatory network of p53-induced apoptosis. A potential impact of the proposed research is to be able to predict cancer-developing processes from the proposed stochastic modeling and analysis, as a result to aid in early cancer diagnosis and prediction.

p53 plays a critical role in the suppression of tumorgenesis. Subsequent studies demonstrated that p53 is a transcription factor with tumor-suppressor activity. Human p53 structural gene is mutated in more than 50% of primary human tumors, including tumors of the gastrointestinal tract. p53 normally acts as a tumor-suppressor gene by inducing genes that can cause cell cycle arrest or apoptosis and also by inhibiting angiogenesis (new blood vessel formation) [2]. Also, p53 may play other roles in the apoptosis network [90]. As reviewed in earlier chapters, a wide variety of formalisms for the inference of genetic regulatory networks has been studied in the literature. We will mainly focus on Baysian network approach. Based on data sources from genomic and proteomic levels, the tentative plan to carry out the p53 apoptosis network study are as follows: exploring the protein-protein interactive network; learning the genomic and proteomic network structure and dynamics of the p53-induced apoptosis in cancer cells; and designing experimental observations to evaluate and refine the model.

11.5.2. Carbon nanotubes in tumor-suppression therapy

Photodynamic therapy (PDT) uses laser light in the spectrum of 630 nm to activate porphyrin compounds (sensitive to this frequency) administered intravenously and collected within tumor sites. The excited porphyrin structure can then transfer its energy to ground-state triplet O_2 which in turn is excited into its highly reactive excited singlet state. This relatively new therapy is quite promising but has some significant limitations.

One of the most significant clinical limitations of PDT with porphyrin compounds is the high-serum protein affinity and the modest differential selectivity of uptake of the photosensitizing dye into tumor sites. Though human tissue shows a greater permittivity to visible light in the range of 600 nm, its absorption of this wavelength is still quite large and greatly attenuates the amount of activating radiation that is actually applied to the porphyrin dye. Clinical limitations here are the amount of light that can be administered selectively to the tumor site without activating porphyrin dye in the peripheral tissue or damaging peripheral tissue with high level of light energy (burning). In addition to these limitations, the side effects of porphyrin compounds (which include hemoptysis and acute anemia) preclude its use as an adjunctive therapy with other chemotherapy regimens, as many common chemotherapeutic agents suppress hematopoiesis which would further exacerbate the anemia/neutropenia/leukopenia.

Our current research uses the relatively inert carbon nanotube structures to induce a cytotoxic radical activity cascade. The research is carried out specifically at the following aspects.

(i) Analyzing the quantum states of nanotubes under certain magnetic or electrical field conditions. Furthermore, we would determine the possibility for the electrons in carbon nanotube structures to transfer energy in such a manner as to excite the ground-state radical (triplet) O_2 into its nonradical excited singlet state. We are also investigating the effect

of phonon energy transfer into other biologically available compounds that have some cytotoxic effects.

(ii) Exploring the means to excite nanotubes remotely to induce a cytotoxic cascade. This would be an incredible medical/biological tool in that magnetic fields are not attenuated to the degree that light radiation is. This would allow the activation of our cytotoxic agent (nanotubes) even in the deepest most inaccessible tumor sites.

(iii) Developing a method to deliver our cytotoxic nanotubes to specific tumor sites. This will be absolutely necessary as there will be no way to form highly localized magnetic fields within an organism. In the presence of our activating magnetic field, all of our nanotubes will become cytotoxic so we will need to ensure that they are localized to the tumor site.

Bibliography

[1] R. T. Greenlee, M. B. Hill-Harmon, T. Murray, and M. Thun, "Cancer statistics," *CA. Cancer J. Clin.*, vol. 51, no. 1, pp. 15–36, 2001.

[2] K. W. Kinzler and B. Vogelstein, *Colorectal Cancer*, McGraw-Hill, New York, NY, USA, 2001.

[3] A. P. Feinberg, *Genomic Imprinting and Cancer*, McGraw-Hill, New York, NY, USA, 2001.

[4] M. O. Hengartner, "The biochemistry of apoptosis," *Nature*, vol. 407, no. 6805, pp. 770–776, 2000.

[5] A. E. Guttmacher and F. S. Collins, "Welcome to the genomic era," *N. Engl. J. Med.*, vol. 349, no. 10, pp. 996–998, 2003.

[6] F. S. Collins, E. D. Green, A. E. Guttmacher, and M. S. Guyer, "A vision for the future of genomics research," *Nature*, vol. 422, no. 6934, pp. 835–847, 2003.

[7] F. S. Collins, M. Morgan, and A. Patrinos, "The human genome project: lessons from large-scale biology," *Science*, vol. 300, no. 5617, pp. 286–290, 2003.

[8] F. S. Collins, A. Patrinos, E. Jordan, A. Chakravarti, R. Gesteland, and L. Walters, "New goals for the U.S. human genome project: 1998–2003," *Science*, vol. 282, no. 5389, pp. 682–689, 1998.

[9] P. Hieter and M. Boguski, "Functional genomics: it's all how you read it," *Science*, vol. 278, no. 5338, pp. 601–602, 1997.

[10] H. P. Harding, Y. Zhang, H. Zeng, et al., "An integrated stress response regulates amino acid metabolism and resistance to oxidative stress," *Mol. Cell.*, vol. 11, no. 3, pp. 619–633, 2003.

[11] M. P. Washburn, A. Koller, G. Oshiro, et al., "Protein pathway and complex clustering of correlated mRNA and protein expression analyses in Saccharomyces cerevisiae," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 6, pp. 3107–3112, 2003.

[12] S. P. Gygi and R. Aebersold, "Absolute quantitation of 2-D protein spots," *Methods Mol. Biol.*, vol. 112, pp. 417–421, 1999.

[13] A. Pandey and M. Mann, "Proteomics to study genes and genomes," *Nature*, vol. 405, no. 6788, pp. 837–846, 2000.

[14] W. A. Tao and R. Aebersold, "Advances in quantitative proteomics via stable isotope tagging and mass spectrometry," *Curr. Opin. Biotechnol.*, vol. 14, no. 1, pp. 110–118, 2003.

[15] S. Gu, S. Pan, E. M. Bradbury, and X. Chen, "Precise peptide sequencing and protein quantification in the human proteome through in vivo lysine-specific mass tagging," *J. Am. Soc. Mass. Spectrom.*, vol. 14, no. 1, pp. 1–7, 2003.

[16] E. P. Diamandis, "Mass spectrometry as a diagnostic and a cancer biomarker discovery tool: opportunities and potential limitations," *Mol. Cell Proteomics*, vol. 3, no. 4, pp. 367–378, 2004.

[17] A. Cho and D. Normile, "Nobel prize in chemistry. Mastering macromolecules," *Science*, vol. 298, no. 5593, pp. 527–528, 2002.

[18] R. Aebersold and M. Mann, "Mass spectrometry-based proteomics," *Nature*, vol. 422, no. 6928, pp. 198–207, 2003.

[19] S. P. Gygi, B. Rist, S. A. Gerber, F. Turecek, M. H. Gelb, and R. Aebersold, "Quantitative analysis of complex protein mixtures using isotope-coded affinity tags," *Nat. Biotechnol*, vol. 17, no. 10, pp. 994–999, 1999.

[20] S. Pan, S. Gu, E. M. Bradbury, and X. Chen, "Single peptide-based protein identification in human proteome through MALDI-TOF MS coupled with amino acids coded mass tagging," *Anal. Chem.*, vol. 75, no. 6, pp. 1316–1324, 2003.

[21] N. L. Anderson, M. Polanski, R. Pieper, et al., "The human plasma proteome: a nonredundant list developed by combination of four separate sources," *Mol. Cell Proteomics*, vol. 3, no. 4, pp. 311–326, 2004.

[22] G. Di Chiro, R. L. DeLaPaz, R. A. Brooks, et al., "Glucose utilization of cerebral gliomas measured by [18F] fluorodeoxyglucose and positron emission tomography," *Neurology*, vol. 32, no. 12, pp. 1323–1329, 1982.

[23] S. P. Robinson, S. J. Barton, P. M. McSheehy, and J. R. Griffiths, "Nuclear magnetic resonance spectroscopy of cancer," *Br. J. Radiol.*, vol. 70, pp. S60–S69, 1997.

[24] E. Olavarria, E. Kanfer, R. Szydlo, et al., "Early detection of BCR-ABL transcripts by quantitative reverse transcriptase-polymerase chain reaction predicts outcome after allogeneic stem cell transplantation for chronic myeloid leukemia," *Blood*, vol. 97, no. 6, pp. 1560–1565, 2001.

[25] E. Schrock, S. du Manoir, T. Veldman, et al., "Multicolor spectral karyotyping of human chromosomes," *Science*, vol. 273, no. 5274, pp. 494–497, 1996.

[26] T. Ried, M. Liyanage, S. du Manoir, et al., "Tumor cytogenetics revisited: comparative genomic hybridization and spectral karyotyping," *J. Mol. Med.*, vol. 75, no. 11-12, pp. 801–814, 1997.

[27] W. A. Bonner, H. R. Hulett, R. G. Sweet, and L. A. Herzenberg, "Fluorescence activated cell sorting," *Rev. Sci. Instrum.*, vol. 43, no. 3, pp. 404–409, 1972.

[28] C. Cooper, "Review: Application of microarray technology in breast cancer research," *Breast Cancer Res.*, vol. 3, pp. 158–175, 2001.

[29] L. Joos, E. Eryuksel, and M. H. Brutsche, "Functional genomics and gene microarrays—the use in research and clinical medicine," *Swiss. Med. Wkly.*, vol. 133, no. 3-4, pp. 31–38, 2003.

[30] M. Gabig and G. Wegrzyn, "An introduction to DNA chips: principles, technology, applications and analysis," *Acta. Biochim. Pol.*, vol. 48, no. 3, pp. 615–622, 2001.

[31] R. J. Lipshutz, S. P. Fodor, T. R. Gingeras, and D. J. Lockhart, "High density synthetic oligonucleotide arrays," *Nat. Genet.*, vol. 21, Suppl 1, pp. 20–24, 1999.

[32] T. R. Hughes, M. Mao, A. R. Jones, et al., "Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer," *Nat. Biotechnol.*, vol. 19, no. 4, pp. 342–347, 2001.

[33] D. G. Wang, J. B. Fan, C. J. Siao, et al., "Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome," *Science*, vol. 280, no. 5366, pp. 1077–1082, 1998.

[34] Y. Takahashi, Y. Ishii, T. Nagata, M. Ikarashi, K. Ishikawa, and S. Asai, "Clinical application of oligonucleotide probe array for full-length gene sequencing of TP53 in colon cancer," *Oncology*, vol. 64, no. 1, pp. 54–60, 2003.

[35] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[36] C. M. Perou, S. S. Jeffrey, M. van de Rijn, et al., "Distinctive gene expression patterns in human mammary epithelial cells and breast cancers," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 16, pp. 9212–9217, 1999.

[37] A. A. Alizadeh, M. B. Eisen, R. E. Davis, et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

[38] U. Alon, N. Barkai, D. A. Notterman, et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 12, pp. 6745–6750, 1999.

[39] C. Debouck and P. N. Goodfellow, "DNA microarrays in drug discovery and development," *Nat. Genet.*, vol. 21, no. 1, pp. 48–50, 1999.

[40] D. V. Nguyen and D. M. Rocke, "Tumor classification by partial least squares using microarray gene expression data," *Bioinformatics*, vol. 18, no. 1, pp. 39–50, 2002.

[41] G. Bloom, I. V. Yang, D. Boulware, et al., "Multi-platform, multi-site, microarray-based human tumor classification," *Am. J. Pathol.*, vol. 164, no. 1, pp. 9–16, 2004.

[42] J. C. Chang, E. C. Wooten, A. Tsimelzon, et al., "Gene expression profiling for the prediction of therapeutic response to docetaxel in patients with breast cancer," *Lancet*, vol. 362, no. 9381, pp. 362–369, 2003.

[43] L. J. van't Veer, H. Dai, M. J. van de Vijver, et al., "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.

[44] U. Scherf, D. T. Ross, M. Waltham, et al., "A gene expression database for the molecular pharmacology of cancer," *Nat. Genet.*, vol. 24, no. 3, pp. 236–244, 2000.

[45] M. J. Marton, J. L. DeRisi, H. A. Bennett, et al., "Drug target validation and identification of secondary drug target effects using DNA microarrays," *Nat. Med.*, vol. 4, no. 11, pp. 1293–1301, 1998.

[46] T. R. Hughes, M. J. Marton, A. R. Jones, et al., "Functional discovery via a compendium of expression profiles," *Cell*, vol. 102, no. 1, pp. 109–126, 2000.

[47] M. Orr, A. Williams, and L. Vogt, *Discovery of 830 Candidate Therapeutic Targets and Diagnostic Markers for Breast Cancer Using Oligonucleotide Microarray Technology*, Nature Publishing Group, London, 2001.

[48] F. S. Collins, M. S. Guyer, and A. Charkravarti, "Variations on a theme: cataloging human DNA sequence variation," *Science*, vol. 278, no. 5343, pp. 1580–1581, 1997.

[49] J. H. Schiller, D. Harrington, C. P. Belani, et al., "Comparison of four chemotherapy regimens for advanced non-small-cell lung cancer," *N. Engl. J. Med.*, vol. 346, no. 2, pp. 92–98, 2002.

[50] N. W. Kim, M. A. Piatyszek, K. R. Prowse, et al., "Specific association of human telomerase activity with immortal cells and cancer," *Science*, vol. 266, no. 5193, pp. 2011–2015, 1994.

[51] D. Kipling, *The Telomere*, Oxford University Press, Oxford, UK, 1995.

[52] K. C. Healy, "Telomere dynamics and telomerase activation in tumor progression: prospects for prognosis and therapy," *Oncol. Res.*, vol. 7, no. 3–4, pp. 121–130, 1995.

[53] W. C. Hahn, S. A. Stewart, M. W. Brooks, et al., "Inhibition of telomerase limits the growth of human cancer cells," *Nat. Med.*, vol. 5, no. 10, pp. 1164–1170, 1999.

[54] C. B. Harley and N. W. Kim, "Telomerase and cancer," in *Important Advances Oncology*, V. T. De Vita, S. Hellmann, and S. A. Rosemberg, Eds., pp. 57–67, Lippincott-Raven, Philadelphia, USA, 1996.

[55] M. M. Kim, M. A. Rivera, I. L. Botchkina, R. Shalaby, A. D. Thor, and E. H. Blackburn, "A low threshold level of expression of mutant-template telomerase RNA inhibits human tumor cell proliferation," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 14, pp. 7982–7987, 2001.

[56] J. F. Kerr, A. H. Wyllie, and A. R. Currie, "Apoptosis: a basic biological phenomenon with wide-ranging implications in tissue kinetics," *Br. J. Cancer*, vol. 26, no. 4, pp. 239–257, 1972.

[57] J. M. Adams and S. Cory, "The Bcl-2 protein family: arbiters of cell survival," *Science*, vol. 281, no. 5381, pp. 1322–1326, 1998.

[58] M. Hollstein, D. Sidransky, B. Vogelstein, and C. C. Harris, "p53 mutations in human cancers," *Science*, vol. 253, no. 5015, pp. 49–53, 1991.

[59] B. Jansen, V. Wacheck, E. Heere-Ress, et al., "Chemosensitisation of malignant melanoma by BCL2 antisense therapy," *Lancet*, vol. 356, no. 9243, pp. 1728–1733, 2000.

[60] S. Singhal, J. Mehta, R. Desikan, et al., "Antitumor activity of thalidomide in refractory multiple myeloma," *N. Engl. J. Med.*, vol. 341, no. 21, pp. 1565–1571, 1999.

[61] H. Hurwitz, L. Fehrenbacher, T. Cartwright, et al., "Bevacizumab (a monoclonal antibody to vascular endothelial growth factor) prolongs survival in first-line colorectal cancer (CRC): results of a phase III trial of bevacizumab in combination with bolus IFL (irinotecan, 5-fluoruracil, leucovorin) as first-line therapy in subjects with metastatic CRC," in *Proc. American Society of Clinical Oncology (ASCO '03)*, vol. 3646, Chicago, Ill, USA, 2003.

[62] D. Cunningham, Y. Humblet, S. Siena, et al., "Cetuximab (C225) alone or in combination with irinotecan (CPT-11) in patients with epidermal growth factor receptor (EGFR)-positive, irinotecan-refractory metastatic colorectal cancer (MCRC)," *Proc. Am. Soc. Clin. Oncol.*, vol. 22, pp. 252, 2003.

[63] P. G. Richardson, B. Barlogie, J. Berenson, et al., "A phase 2 study of bortezomib in relapsed, refractory myeloma," *N. Engl. J. Med.*, vol. 348, no. 26, pp. 2609–2617, 2003.

[64] P. G. Richardson, J. Berenson, D. Irwin, et al., "Phase II study of PS-341, a novel proteasome inhibitor, alone or in combination with dexamethasone in patients with multiple myeloma who have relapsed following front-line therapy and are refractory to their most recent therapy," *Blood*, vol. 98, no. 11, pp. 3223, 2001.

[65] H. Lemke, G. J. Hammerling, C. Hohmann, and K. Rajewsky, "Hybrid cell lines secreting monoclonal antibody specific for major histocompatibility antigens of the mouse," *Nature*, vol. 271, no. 5642, pp. 249–251, 1978.

[66] S. G. O'Brien, F. Guilhot, R. A. Larson, et al., "Imatinib compared with interferon and low-dose cytarabine for newly diagnosed chronic-phase chronic myeloid leukemia," *N. Engl. J. Med.*, vol. 348, no. 11, pp. 994–1004, 2003.

[67] H. Joensuu, P. J. Roberts, M. Sarlomo-Rikala, et al., "Effect of the tyrosine kinase inhibitor STI571 in a patient with a metastatic gastrointestinal stromal tumor," *N. Engl. J. Med.*, vol. 344, no. 14, pp. 1052–1056, 2001.

[68] G. D. Demetri, M. von Mehren, C. D. Blanke, et al., "Efficacy and safety of imatinib mesylate in advanced gastrointestinal stromal tumors," *N. Engl. J. Med.*, vol. 347, no. 7, pp. 472–480, 2002.

[69] M. Fukuoka, S. Yano, G. Giaccone, et al., "Multi-institutional randomized phase II trial of gefitinib for previously treated patients with advanced non-small-cell lung cancer," *J. Clin. Oncol.*, vol. 21, no. 12, pp. 2237–2246, 2003.

[70] M. G. Kris, R. B. Natale, R. S. Herbst, et al., "Efficacy of Gefitinib, an inhibitor of the epidermal growth factor receptor tyrosine kinase, in symptomatic patients with nonsmall cell lung cancer," *JAMA*, vol. 290, no. 16, pp. 2149–2158, 2003.

[71] T. J. Lynch, D. W. Bell, R. Sordella, et al., "Activating mutations in the epidermal growth factor receptor underlying responsiveness of non-small-cell lung cancer to gefitinib," *N. Engl. J. Med.*, vol. 350, no. 21, pp. 2129–2139, 2004.

[72] G. A. Wiseman, L. I. Gordon, P. S. Multani, et al., "Ibritumomab tiuxetan radioimmunotherapy for patients with relapsed or refractory non-Hodgkin lymphoma and mild thrombocytopenia: a phase II multicenter trial," *Blood*, vol. 99, no. 12, pp. 4336–4342, 2002.

[73] M. S. Kaminski, K. R. Zasadny, I. R. Francis, et al., "Radioimmunotherapy of B-cell lymphoma with [131I]anti-B1 (anti-CD20) antibody," *N. Engl. J. Med.*, vol. 329, no. 7, pp. 459–465, 1993.

[74] M. S. Kaminski, A. D. Zelenetz, O. W. Press, et al., "Pivotal study of iodine I 131 tositumomab for chemotherapy-refractory low-grade or transformed low-grade B-cell non-Hodgkin's lymphomas," *J. Clin. Oncol.*, vol. 19, no. 19, pp. 3918–3928, 2001.

[75] A. Dickson and S. Calderwood, *Thermosensitivity of Neoplastic Tissues in Vivo. Hyperthermia in Cancer Therapy*, K. Storm, ed., GK Hall, Boston, Mass, USA, 1983.

[76] S. E. Lyshevski, *MEMS and NEMS: Systems, Devices, and Structures*, CRC Press, Boca Raton, Fla, USA, 2002.

[77] R. A. Freitas Jr., *Nanomedicine, Vol. I: Basic Capabilities*, Landes Bioscience, Georgetown, Tex, USA, 1999.

[78] S. Iijima, "Helical microtubules of graphitic carbon," *Nature*, vol. 354, no. 6348, pp. 56–58, 1991.

[79] M. A. Reed, J. Chen, A. M. Rawlett, D. W. Price, and J. M. Tour, "Molecular random access memory cell," *Applied Physics Letters*, vol. 78, no. 23, pp. 3735–3737, 2001.

[80] C. Zhou, M. R. Deshpande, M. A. Reed, L. Jones II, and Tour J. M., "Nanoscale metal/self-assembled monolayer/metal heterostructures," *Applied Physics Letters*, vol. 71, no. 5, pp. 611–613, 1997.

[81] S. Datta, *Electronic Transport in Mesoscopic Systems*, Cambridge University Press, Cambridge, UK, 1995.

[82] R. F. Curl and R. E. Smalley, "Probing C_{60}," *Science*, vol. 242, no. 4881, pp. 1017–1022, 1988.

[83] Z. Zhu, D. I. Schuster, and M. E. Tuckerman, "Molecular dynamics study of the connection between flap closing and binding of fullerene-based inhibitors of the HIV-1 protease," *Biochemistry*, vol. 42, no. 5, pp. 1326–1333, 2003.

[84] S. J. Wind, J. Appenzeller, and P. Avouris, "Lateral scaling in carbon-nanotube field-effect transistors," *Phys. Rev. Lett.*, vol. 91, no. 5, pp. 058301, 2003.

[85] J. W. G. Wildoer, L. C. Venema, A. G. Rinzler, R. E. Smalley, and C. Dekker, "Electronic structure of atomically resolved carbon nanotubes," *Nature*, vol. 391, no. 59, pp. 59–62, 1998.
[86] T. W. Odom, J. Huang, P. Kim, and C. M. Lieber, "Atomic structure and electronic properties of single-walled carbon nanotubes," *Nature*, vol. 391, pp. 62–64, 1998.
[87] S. Frank, P. Poncharal, Z. L. Wang, and W. A. Heer, "Carbon nanotube quantum resistors," *Science*, vol. 280, no. 5370, pp. 1744–1746, 1998.
[88] G. Stix, "Little big science. Nanotechnology," *Sci. Am.*, vol. 285, no. 3, pp. 32–37, 2001.
[89] M. G. Harisinghani, J. Barentsz, P. F. Hahn, et al., "Noninvasive detection of clinically occult lymph-node metastases in prostate cancer," *N. Engl. J. Med.*, vol. 348, no. 25, pp. 2491–2499, 2003.
[90] G. M. Wahl and A. M. Carr, "The evolution of diverse biological responses to DNA damage: insights from yeast and p53," *Nat. Cell. Biol.*, vol. 3, no. 12, pp. E277–E286, 2001.
[91] A. Meller, L. Nivon, and D. Branton, "Voltage-driven DNA translocations through a nanopore," *Phys. Rev. Lett.*, vol. 86, no. 15, pp. 3435–3438, 2001.

X. Steve Fu: The Corvallis Clinic, 3680 NW Samaritan Dr., Corvallis, OR 97330, USA
Email: steve.fu@corvallis-clinic.com

Chien-an A. Hu: School of Medicine, University of New Mexico, Albuquerque, NM 87131, USA
Email: ahu@salud.unm.edu

Jie Chen: Division of Engineering, Brown University, Providence, RI 02912, USA
Email: jie_chen@brown.edu

Z. Jane Wang: Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada
Email: zjanew@eee.ubc.ca

K. J. Ray Liu: Communication and Signal Processing Laboratory, Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA
Email: kjrliu@eng.umd.edu

12 Integrated approach for computational systems biology

Seungchan Kim, Phillip Stafford,
Michael L. Bittner, and Edward B. Suh

12.1. Background

New technological advancements for the measurement of biological systems have given us much insight into genomic, transcriptomic, and proteomic views of a cell's behavior. Such recent advancements in the measurement technology include expression arrays [1], single nucleotide polymorphism (SNP) [2, 3], CpG island arrays [4], protein abundance and specialized glycoarrays [5, 6], and siRNA [7, 8, 9]. Different measurement techniques are meant to provide different kinds and resolutions of the information regarding target biological systems; therefore, choosing appropriate measurements for a given biological problem is considered fundamental in the solution of the problem. In addition to the technologies that provide a unique snapshot of different aspects of the cellular milieu, we now have the computational and data management challenge of storing, integrating, and analyzing data independently and when mixed. Data storage techniques become increasingly important when integration, and analysis are needed. Database design and planning are now as important as the analysis technologies that are being developed.

Biological problems of special importance now include the recognition of disease subtypes, identification of molecular markers for certain disease types, inference of regulatory mechanisms, discovery of new therapeutic targets for intervention and treatment of disease progression, and the development of novel single and additive drugs and therapeutics. Since the beginning of the modern biological era, the importance and applicability of mathematical, statistical and engineering tools has become quite clear. The Human Genome Project is a primary example. Numerous pattern recognition techniques have been applied to identify molecular markers for a specific disease as well as the identification of disease subtypes. Machine learning and Bayesian frameworks have proven to be effective in learning the mechanisms of genetic regulatory networks, and control theory is being applied to derive a better approach to therapeutic design. As the complexity of biological data increases, it is the combination, not a single specialized tool, which will be most efficacious to solving complex biological problems.

By taking an integrated approach to these biological problems, the *systems biology* dogma strives to bring in technologies from other disciplines and/or to develop truly novel methodologies to find answers to critical biological questions [10, 11, 12]. It is designed to view biology not just at the component-level (e.g., genes and proteins), but also in its behavior at the system-wide level. For example, the study of the robustness of a biological system with regard to its intrinsic and extrinsic noise has received much attention recently [13, 14]. In general, systems biology is described by:

(1) the integration of discovery science and hypothesis-driven science,
(2) the use of biology as an informational science,
(3) the utilization of powerful new high-throughput tools for systematically perturbing and monitoring biological systems,
(4) the creation of new computational methods for modeling and analysis.

This approach is different from traditional biological science which focuses on localized events such as interactions between a small number of genes or proteins and on their detailed biochemical analysis. The holistic view that a biological system is alive and its behavior is not simply explained by the sum of the behaviors of the individual parts has started to take off in biology. Since a system's view on biology becomes more and more important, so does mathematical and computation modeling of biological systems in systems biology. Therefore, more and more research institutes put together multidisciplinary groups of scientists, such as molecular biologists, mathematicians, physicists, computer scientists, and engineers to name a few.

Systems biology considers the integration of both experimental and analytical approaches. For example, researchers can use microarray data to grasp a genomic view of tumors and identify candidate molecular signatures of the tumor. They can also use array Comparative Genomic Hybridization (CGH) [15, 16] to see if the differential expression is due to copy number variation, methylation arrays to determine if CpG islands are differentially methylated, protein arrays to determine protein abundance and correlation to mRNA abundance numbers, and specialized protein arrays such as glycoarrays to determine the differential post-translational modifications that affect many proteins. Different types of data require different types of analyses and interpretation of the results. Therefore, the experiments with multitype measurements call for integrated suite of analytical tools. Also, another kind of information critical in modern biomedical research is knowledge information and its integration with data-driven analysis. Recently, a significant number of studies and publications employ such an integrated approach. Therefore, the advancement of systems biology that integrates multitype data, knowledge, and analytical tools is critical for biomedical research.

Figure 12.1 is an overview of systems biology with an emphasis on modeling and analysis. It is broken into three panels. The top panel concerns data mining and pattern analysis of genomic and proteomic data. Data mining and pattern

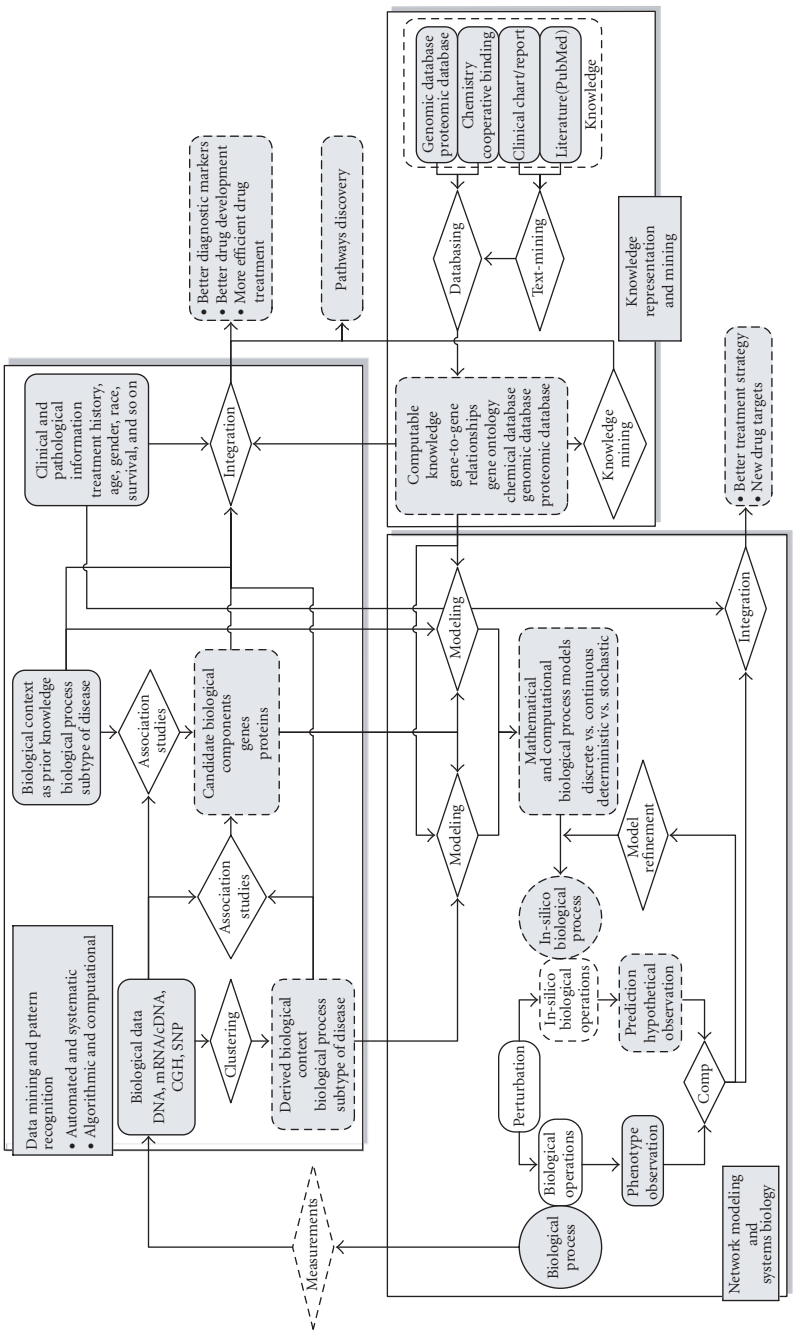


Figure 12.1. Overview of modeling and analysis flow for systems biology.

analysis employ various statistical learning methods such as clustering, classification, rule learning to find biological contexts, molecular markers associated with a specific context, and rules to predict clinical or prognostic outcomes, for example, in diseases. The bottom left panel deals with mathematical models for a biological system (synthesis) and analysis of the models to predict the behavior of a biological process and generate new hypotheses for further validation. Models developed in a study may require the development of a new measurement technology or call for a new design of biological experiments. Another critical component is depicted in the bottom right panel. As biological knowledge is accumulated, the utilization of known knowledge (the "knowledgebase") plays a critical role in understanding the biological system and the discovery and integration of new knowledge. Various text-mining and content-mining resources should be developed and deployed to further exploit this vast amount of data waiting to be analyzed. In addition, artificial intelligence methods (knowledge reasoning) will play an increasingly pivotal role in the extraction of new knowledge.

In this chapter, we briefly introduce some of the analytical tools that have proven to be useful in the study of systems biology and show a few examples of biological studies using such tools in the context of systems biology. We will address the computational tools currently available and actively being used for the study of cellular systems and present an example of the integrative use of such tools to answer questions. However, we should first consider briefly what kinds of measurements are available enabling the study of computational systems biology.

While systems biology tends to focus on mathematical modeling of regulatory networks and signal transduction networks, which is of utmost interest to biologists, a systems biology approach can be also applied to different problems. A clinical study described by Kitano [17] is one such useful example, and it guides us to look at the problem in various perspectives before deploying more complicated tools to model the system mathematically. Figure 12.2 describes well the analogy between clinical studies and systems biology study for gene regulatory networks. Both have repeated feedbacks between hypothesis development/synthesis and its testing in the process of discovery of new knowledge.

Among some of the mathematical, computational, and statistical tools that are actively used in systems biology are pattern identification for the subtypes of diseases and the identification of associated genes [1, 18, 19, 20, 21, 22, 23, 24], prediction modeling to find functional relationships among genes and/or proteins given the cellular context [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47], and mathematical and computational modeling of gene regulatory networks to study how genes and proteins are interacting with each other to generate a certain phenotype either in disease or in response to a specific treatment. Recent studies on network modeling emphasizes whether a proposed model exhibits robustness [37, 48, 49, 50, 51, 52, 53] and scale-free characteristics [54, 55, 56], which are believed to be needed in genetic regulatory networks for maintaining homeostasis [57] as well as for coping with certain levels of intrinsic and extrinsic uncertainty in biological systems [58, 59]. Another component of systems biology, recently emerged, is knowledge dissemination/integration to

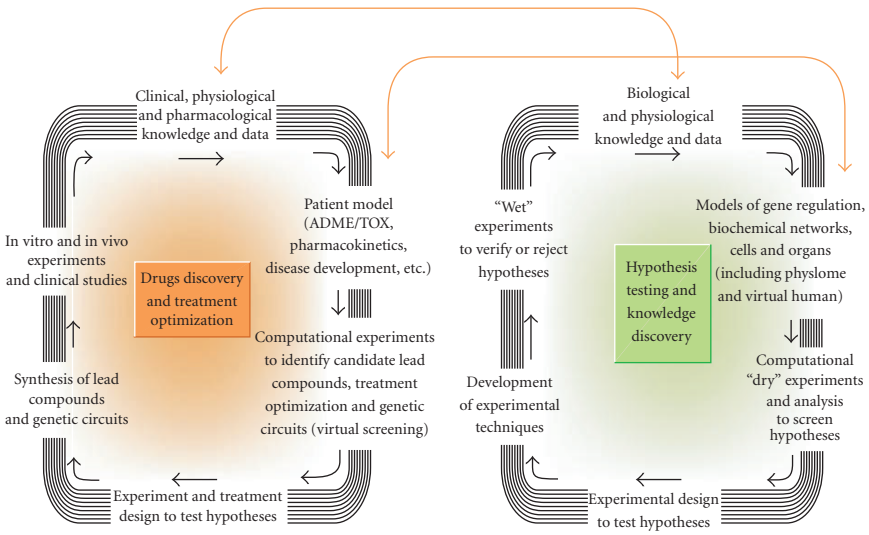


Figure 12.2. Analogy between clinical studies and basic systems biology research. Systems biology is an integrated process of computational modeling, system analysis, technology development for experiments, and quantitative experiments. With sufficient progress in basic systems biology, this cycle can be applied to drug discovery and the development of new treatments. In the future, *in silico* experiments and screening of lead candidates and multiple drug systems, as well as introduced genetic circuits, will have a key role in the "upstream" processes of the pharmaceutical industry, significantly reducing costs and increasing the success of product and service development [17].

accelerate new hypotheses based on new observations as well as prior knowledge. Content- and text-mining tools should play a key role in this, in conjunction with knowledge reasoning tools from artificial intelligence systems.

12.2. Biological data and measurement technologies

Understanding what information at a particular level of resolution we should expect from observation technologies (e.g., expression profiling) is the first step in the study of systems biology. Data availability is, at a very basic level, necessary for successfully modeling a biological process. No truly robust biological model could be created using information from a single view of biological activity. In essence, a transcriptional profile merely indicates *potential* downstream translational activity. A protein abundance profile merely indicates the presence of proteins, but does not indicate posttranslational modifications or targeting. Knowing all posttranslational modifications will not necessarily indicate all possible protein-protein interactions, and so on. The point that the presence of sufficient information about certain molecular events is not inherently sufficient. As mentioned above, it is necessary to collect information in layers and it is the interaction, integration, and analysis of these layers that truly fulfils the principle dogma of systems biology.

Ideally, a biologist would like to capture a complex biological process mathematically using overlapping views of a cell along the axis that defines the observational technology (gene expression, genomic sequence, protein abundance, interfering RNA, synthetic lethals or gene-gene interactions, etc.) and perpendicularly along the time axis. Biology can be considered as fundamentally complex as weather patterns, but, as opposed to weather phenomena, cells can be captured, isolated, perturbed, and observed. Importantly, our selected observational axis must provide adequate views to overlap and validate every other view and there must be sufficient resolution to accommodate the biological and technical variability without excessive false positives. This overlap can be exemplified by looking at a situation where we have SNP data, promoter location information, transcription factor information, expression data, and protein abundance data. Each of these individual observations is a view of the cell from a certain perspective and each reports information about a particular function (along with certain inherent biases), but none can be extended enough to completely explain all of the factors that cause a cell to survive and maintain a healthy equilibrium. With enough different types of data, we can validate one data type with another. This may seem contraindicated since we know that transcriptional profiles do not match protein abundance very well; typically only 50% of the transcripts in a cell correlate with protein abundance, however this apparent discrepancy can be accommodated somewhat by knowing something about the stability of the molecular species in question. Thus, one extra layer of information helps to reconcile two separate seemingly controvertible observations. The observational technologies provide the "angles," or dimensions, with which to view the biological state, and user knowledge ties together the bits of information seamlessly. Two types of observations are possible at this point: a view of the cell in an unperturbed or terminal phenotype (e.g., at some stationary point along the cell cycle) or a perturbation study where data is collected along a time course as the cell returns to equilibrium.

Using the aforementioned example, we have yet another set of choices to make in our observations. We can collect data that is relatively stable (genomic sequence, SNPs, synthetic lethal screens) or highly dynamic (expression, protein abundance, chromosome amplification/deletion, CpG island methylation, etc., many of which are extremely prevalent in most forms of cancer and many other genetic-based disorders [60]. Experimental design defines the frequency and resolution of data collection and potentially limits the types of analyses that can be performed on the captured data. Although highly defined experiments are the easiest to analyze statistically and they provide support for the proposed hypotheses, data-mining experiments are often quite useful in their own right for extending analyses beyond the original boundaries of the experiment. Interestingly, these types of experiments are often quite underpowered because they attempt to provide too much insight with too little data. While we may not have measurement technologies that allow us to observe biological systems at our desired level of details, the current level of molecular observational technology still enables us to visualize highly complex and detailed aspects of the physiology of a cell or a living system. The frustration of systems biologists stems from the lack of a firm endpoint where

one can create a biological model that truly defines most or all of the parameters of a biological process, and so we extend the hypotheses we create using whatever limited observations we have access to in order to try to explain phenomena that actually require much more details.

As the genomic sequences for more species become available, the blueprint about basic components that drive living cellular systems becomes increasingly clear. The basis for our genomic knowledge is very useful because it works as a reference (normality) against abnormality, that is, mutations that lead to genetic diseases. SNPs can be used to detect the aberration from reference that might indicate a disease basis or susceptibility. Several commercial platforms are currently available that measure genomes for SNPs at resolutions approaching 20 kb and smaller. Now SNP data from populations with heritable disease phenotypes can be interrogated to locate those patterns of polymorphisms that are strongly associated with the phenotype (including but not limited to complex traits such as disease susceptibility, disease progression, late-onset or environmentally influenced epigenetic effects).

At the heart of SNP measurements is the correlation of sequence polymorphisms to phenotype. This correlation is often obtained through a tedious process of identifying as many markers as possible with high coverage and validating those genes and gene products contained within the implicated region(s) by the genetic markers. We use knowledge about the interrelatedness of SNPs, transcription and translation effects, and protein interactions in order to structure and query our data to create the insights that we are seeking, and it is this integration of data *prior* to any analysis that is noteworthy and quite challenging. It is of course possible to analyze each of the aforementioned datasets independently, with data-driven hypotheses, and an attempt to combine the results to support or reject a global hypothesis, but systems biology is making a concerted effort to combine as many types of information about a cell as possible *before analysis actually begins*.

SNPs are found approximately every 2 kb in the genome and can be found in coding regions, within introns or exons, within or surrounding promoter regions, enhancers, 3′ UTR's, and so forth. The driving question is whether a particular SNP is merely a silent marker with no obvious discernible function, or a polymorphism that alters the expression or translation of a particular gene. SNPs occur only in genomic DNA but they affect all aspects of posttranscriptional behavior. In DNA, SNPs affect replication, DNA-protein interactions, local DNA structure, supercoiling and/or DNA stability. During transcription, SNPs may effect initiation, proofreading, elongation, or termination. During mRNA processing, SNPs may alter the binding efficiency of a transcription factor [61], modification of a splice site location [62], or the efficiency of splicesome binding [63] or even polyadenylation [64]. In addition, SNPs may change the amino acid used in the translated product, the glycosylation pattern, proteolytic cleavage, organelle targeting, acylation, methylation, phosphorylation, prenylation, or other posttranslational modifications. At the DNA level, genomic DNA interacts to form supercoiled helixes and interactions with histones. SNPs can alter these interactions causing local changes

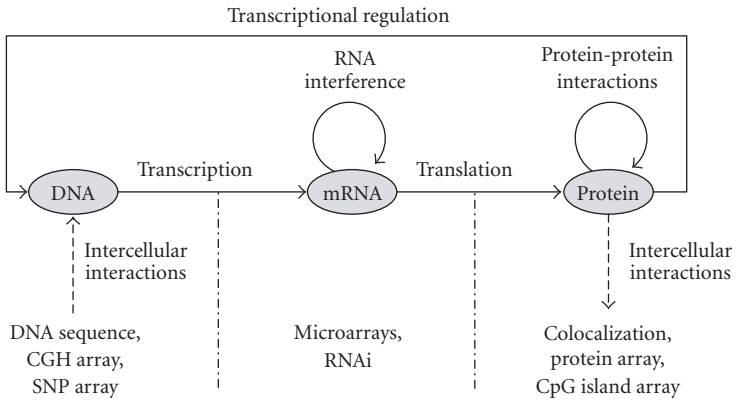


Figure 12.3. Measurement and perturbation technologies at various levels/stages of cellular systems.

that may have pleiotropic effects, and SNPs can even be tissue specific causing epigenetic effects, as in colorectal cancer [65]. As you can see, association is simple through calculations of linkage disequilibrium, loss of heterozygosity, LOD scores, and so forth, but causation is extremely difficult to validate. Knowing the many layers that SNPs can affect gives one the perspective that is required for combining data from multiple measurement platforms into a structure that can be queried, integrated, and eventually used for hypothesis testing and model building. In the above case, candidate SNPs that affect transcription or translated protein structure are validated through the actual integration process itself.

The most popular measurement system used in the application of systems biology to biological questions is the gene expression microarray. It measures transcriptional activities of tens of thousands of genes simultaneously, resulting in individual snapshots of the transcriptional state of a cell's transcriptional image at any given time. While it reflects the dynamic processes of a biological system, it fails to capture various critical aspects of the cell's entire system such as possibly poor correlation between transcriptional level and protein level and posttranslational modifications. To compensate for this, other measurement technologies, that is, protein abundance and interaction arrays [5, 6], can be combined with expression data to get a matched transcriptional/translational profile.

Another important recent advancement is to use siRNA to knock down the activity of mRNA temporarily in order to perturb the target biological system and observe its response as it returns to equilibrium. In conjunction with tissue and cell arrays for the measurement of localizable events in cells, siRNA technology provides us with an efficient way to systematically perturb biological systems and monitor their response to perturbation [9, 66]. The number of mRNAs that can be knocked down and visualized through cellular imaging is increasing rapidly and genome-wide siRNA libraries will be available soon. Figure 12.3 summarizes some of the measurement techniques and perturbation methods available these days at various levels and stages of cellular behavior.

Recently, highly detailed measurements depicting molecular and biochemical activity in individual cells have become available—these details enhance the development of much more detailed mathematical modeling and validation [67, 68, 69, 70]. The number of biological components that can be measured simultaneously in an individual cell is increasing dramatically by deploying such technologies as fiber-optic microimaging [71, 72, 73].

12.3. System for biological data integration

Once we have captured multiple types of biological data, we have a source that we can build upon for future analyses. Storing and retrieving data is one key to successful analysis of individual experiments and the expansion of analyses beyond the confines of a single data type. Combining data types can make the original data much more useful than the investigator had originally intended. Well-designed relational databases are critical to the successful integration of biological data, and well-designed data structures are critical to the design of databases. In order to model a biological process, one needs sufficient data sources to overcome the inherent limitation in resolution for each of the measurement technologies used. For example, transcriptional profiles are useful for measuring gene product potential, but a protein abundance array is necessary to make the connection between potential and protein. Keep in mind that not all splice variants are measured on an expression array and not all measurements indicate the precise mRNA transcript number in the cell (i.e., the fluorescence intensity per spot is not directly related to the mRNA number). Relational databases are truly useful for linking biologically relevant data from one type of observation (e.g., expression data) with another (e.g., protein abundance) using entity relationships, constrained vocabularies, and well-established principles of data queries. One problem with biological data is the lack of a thorough metathesaurus that defines medical conditions in such a way that disallows free text descriptions (difficult to query) but will still encompass the dynamic range and information content of the data to facilitate successful queries (difficult to capture sufficient information).

In the most common example, we establish a microarray database, that is, MIAME (minimum information about a microarray experiment available at http://www.mged.org/Workgroups/MIAME/miame.html) compliant through the use of the MAGE (microarray and gene expression available at http://www.mged.org/Workgroups/MAGE/mage.html) object model and the MAGE ontology, as well as the inherent hierarchical relationships between and among the data. In this case, MIAME compliance means that the data includes all of the technical details (metadata) for the microarray platform, the experimental design, and the protocols wrapped together and linked within a highly constrained and well-defined XML file. The MIAME experimental details are necessary in order to link this experiment to other types of biological experiments (e.g., clinical data, toxicology measurements, etc.) and to integrate public data with in-house data. MIAME compliance in this case also means that the recipient of the XML file can parse the data without a key or DTD (document type definition), can store the data in a

compliant database, and can recreate the experiment with no further interaction from the original scientist. The requirements are strict but are important in ensuring uniformity, standardization, and ease of data sharing. These requirements are met through the use of a well-defined object model (MAGE-OM) which subsequently facilitates the creation of a formal XML document and a structured data schema that allows simple input and output of XML-based data.

Forms are an important feature of the data capture process. Forms allow flexibility, rational web-based data input pages, and simple design and normalization of data tables. Using biological vocabularies to build forms for data input and submission to public repositories prevents loss of information contained within highly descriptive fields. For example, molecular biologists may recognize Myosin V mutations as a cause of albinism or hypopigmentation, while medical doctors may recognize Chediak-Higashi syndrome and others might know Griscelli's disease. Forms and constrained vocabulary promote queries across data and annotations that might normally be of limited utility. We might wish to associate SNP data with a transcriptional/translational experiment (gene expression and protein abundance) but without some form of text conversion or normalization, we might use different terminology that another scientist has used for a common genetic disease, and we could completely miss highly correlated data. In the MIAME requirements we can always find annotations about the probes that were used to measure expression information. In SNP data, the minimum information about a genotyping experiment (MIAGE) requirements force the user to annotate the dataset so sufficiently that we can correlate the sequence where the SNPs were found with the genes that were interrogated in an expression experiment. To continue our query beyond SNP and expression information, we may also have the patient name and history, clinical information, and all relevant personal descriptors necessary to link a disease or genetic predisposition to illness to SNPs contained within a gene or genes which may have been measured in the expression experiment. For certain epidemiology studies, we may even have extensive blood work or biopsy data which provides even more information to annotate the molecular data we have gathered. It is important to realize that these are separate and distinct databases that have no inherent relationship to one another, nor were they designed to—epidemiology databases are principally created to support a specific and one-time experiment (although the experiment may be continuous) and typically are designed to last for the duration of the project while expression and SNP databases are designed to be added continuously. However, when we do a query across a suitable database federation, we can search for patient name or ID and all relevant information from that epidemiology or clinical information database. If the microarray and SNP experiments were designed to be run at the same time, or even if there were certain patients within the epidemiology database that had expression or SNP follow-up experiments conducted years later, the experimenter will have captured extensive information that enhances the ability to link the molecular and clinical information together. Let us describe a query in detail.

A user has completed a genotyping experiment of 100 patients in an association study of systemic lupus. A 100K SNP chip was used to gather genotypes

for these 100 patients. Human genome build 34 was used as the basis for assigning dbSNP positions and names. Simultaneously, the scientist has taken biopsies from connective tissue, kidney, and the anterior lobe of the brain of deceased lupus patients who showed some form of cerebral edema. The lobes were used to generate RNA for expression arrays, proteins for a protein abundance microarray as well as some specialized biochemistry, and DNA for sequencing, SNP analysis, and CGH arrays. The experimenter ensured that each experiment was fully annotated with all necessary experimental details, and all of those details were stored in a type-specialized database. Now, the statistician doing the analysis knows that he needs to extract several types of data from the federation of databases physically spread across multiple facilities. He starts his queries looking for a patient name and all available records associated with that patient in the context of lupus. In the system described in Figure 12.4 we can see that the query will return the database names and instances that contain the patient histories conforming to his lupus query. The user sees that databases named Clinical Trials, Patient History, Proteomics, Microarray Gene Expression, CGH, and SNP all contain the patient name and a reference to the particular disease the user is looking for. He discovers that at least 100 patient names appear in four separate experiments (expression, protein abundance, CGH, SNP). The user extracts all of the Patient History information for the patients names and finds that at least one of the patients has been a long-time sufferer of lupus and has been given drugs and autoimmune therapy for at least 1 year prior to death. The Clinical Trials Information database shows that 50 patients have been involved in several phase-I and phase-II trials with experimental drugs that had little or no long-term positive effects on the patients' health. Finally, the SNP data indicated that 20 patients were in a case-control study several months prior to death, and the proteomic, expression, and CGH databases showed that there were 15 patients where a lobectomy was performed immediately upon death with the tissues subsequently being used for expression profiling, protein abundance, and CGH arrays.

The analyst now has the option of querying the databases using newly found information that makes the query much more specific. The analyst downloads information pertaining to individual experiments, and that information leads to new information that could provide public or supplemental data in support of the current project. The user now applies the metadata and metathesaurus database in order to create a smart query. The user wants to filter the protein and expression data somewhat in order to encompass only those regions that were identified from a linkage disequilibrium calculation using the SNP data. The analyst has identified the genomic region for the particular chromosome that he knows is important to lupus susceptibility. That region is not meaningful to the analysis of expression array data because mRNA and protein abundance analysis do not directly use physical locations, but sequence and locations information about a particular expression probe is always available. The analyst would likely want to see the chromosome map annotated with the location of all relevant SNPs in that region. The physical position on the chromosome can be used to extract gene symbols that can then be used to find the protein names that were used on the protein chip.

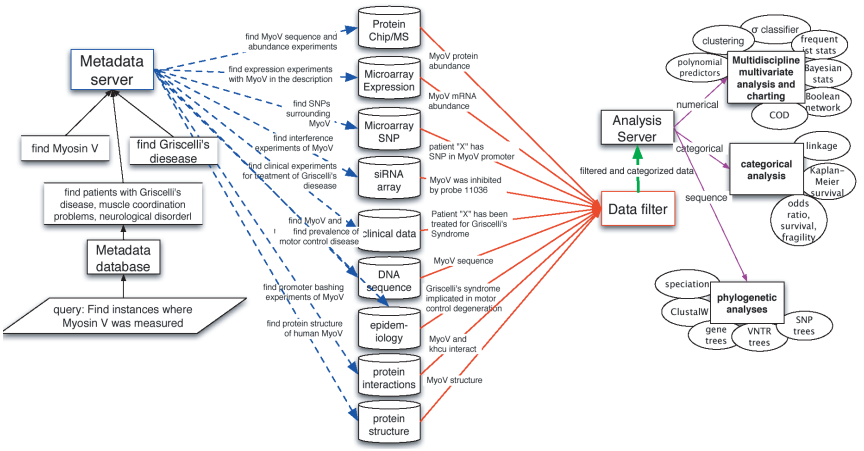


Figure 12.4. Description of a federated system of databases for systems biology. The left diagram (dotted lines) shows a putative query for Myosin V and associated disease information. The lines on the right show the reworked query results that provide much more consistency and usability for statistical analyses. The right side also highlights the data receipt modules that shuttle data into and out of analysis routines, which are split into 3 distinct groups—categorical (clinical or patient outcome, epidemiology, disease association), genetic and phylogenetic (multiple genome comparisons and evolution), and numerical data-mining Bayesian and frequency statistics, classification, neural networks, support vector machine (SVM, etc.). These analysis tools are used to interrogate as much of the data as possible in native format, group by group, and then to analyze all of the data that can be combined into one universal type (i.e., binned, or quantized values).

The gene symbols are used to identify the accession numbers that then link to probe names that correspond to the expression microarray. Armed with a little prior knowledge about the SNP experiment, the user can now use this federated system to extract only the information he needs to do his analysis. This is very different from having to download all of the data manually, and filter only the information necessary for a particular analysis. Ideally, all of this can be done at the database query stage.

Let us go through the actual content of a data management and analysis system that can be set up at any institution with any software and hardware base. In Figure 12.4 we see a series of databases (center) and a set of queries going to the databases and the results coming out. We see a query from a user going into the metadata server that actively queries the metadata database. The metadata database contains all of the possible terms and semantic references that might exist within the biological domain pertaining to the user's query (i.e., NCI's metathesaurus browser located at http://ncimeta.nci.nih.gov/indexMetaphrase.html). This database examines key words in the query and makes inferences based on the context. For example, given the use of the words MyoV, Griscelli's disease and clinical data, the server would look for all synonyms of MyoV including Myosin V, actin-based motors, Myo5, Myo5a, and so forth. This would encompass gene names that extend beyond humans and into mammals, insects, fungi, and so on.

This covers a large portion of the molecular biological databases that might contain gene sequence information, protein sequences, domains, enzymatic activity, and 3D structures. Again, the key words also contain "disease" and "Griscelli's" so the search engine would include medical terms using the unified medical language system (UMLS, available at http://www.nlm.nih.gov/research/umls/), medical subject heading (MeSH, available at http://www.nlm.nih.gov/mesh/meshhome.html), and other medical/clinical dictionaries. Any cross-references found there would be presented to the user as a possible synonym or extended search parameter, such as "hypopigmentism immune deficiency disease," "partial albinism," "Chediak-Higashi-like syndrome," or "neurological disorder." Once the user has identified the additional terms that he is comfortable with, the system begins its systematic search for those terms. As you can see from Figure 12.4, the queries are assembled and worded differently depending on which database is being queried, however the meaning of the user's original query is preserved and extended to encompass other data types. The queries return data to a filtering system that groups the data based on the data itself, with no previous knowledge of what is being returned. The data is grouped and managed by the experimental information contained within the associated metadata for each data type (i.e., CGH, sequence, expression, SNP, etc.). There is a MIAME-like experimental annotation requirement method for each data type, that is, contained in this federation, so that experimental details about each experiment are enforced, and technical information about the experimental technologies is also referenced. This metagrouping is important not only for grouping data in the holding area (the data filter in Figure 12.4) but is critical for passing along relevant information to the analysis server.

The analytical components of this system must be able to read several distinct types of data formats. In the example in Figure 12.4 we see that there is a large amount of numerical data including mRNA abundance, protein abundance, siRNA array data, and CGH array data. Some clinical data would include a graded response to a drug, survival data in the form of percentage mortality or morbidity, and other scalar data mixed together with categorical data. Prior to the analysis, CGH and SNP data move from scalar values to a category of present or absent, number of chromosomal copies, or markers. SNP data is often read as a numerical value that indicates the amount of DNA hybridizing to a probe. CGH data is read as the number of copies of a chromosomal region and array CGH provides the quantity of genes and genomic elements that are overrepresented in the cell when an amplification event occurs. siRNA data is often read as a phenotype based on a fluorescent marker within a cell, and the data often contains total intensity of fluorescence, structural information (nuclear localization, cytoskeletal, cortical, etc.), and variability of transfection efficiency. Unless the user wishes to pursue advanced quality control methods, these values would be the most useful in a multidisciplinary analysis once they are converted to the form where they would make the most sense. Epidemiology, clinical data, protein structure, protein-protein interactions, synthetic lethal interactions, and other types of categorical data would be sent to the analysis server as

quantized levels where all of the responses are known ahead of time and normalized throughout the expected range. The responses might be slightly different depending on the quality of the data and how much the data has been scrubbed before entering the database. The data filter must convert similar responses into a bin of precise responses for the software to be able to recognize that the data is the same across experiments or across treatments. This is the only way that one can combine experiments that would report the same answer. Unfortunately this method eliminates some of the details inherent in scalar data, in exchange for cross-compatibility, but the trade-off provides more benefit as more data types are added. Eventually the data must be represented in the same way so that an algorithm can perform the same statistical tasks on all types of biological data. Thus defining the scale of an observation, splitting or combining observations to describe a quantizable value, and exploring the amount of information that each observation can contribute (its inherent value) to a final analysis is key to data integration.

Now that the analysis server has the data filtered and stored in a temporary data mart, the analysis tools have an easier time recognizing what software goes with what data. Each data type is clearly defined by type, that is, SNP, expression, clinical history or pathology, and so forth. The analysis server then groups the data headings on the screen for the user to examine. The server suggests one or more of each category of analysis tool for the initial analysis (i.e., *t* test, chi-square, survival model, regression, etc.). The user may choose to obtain initial results of clustering for microarray expression data, linkage for SNP data, phylogenetic trees for sequence data and would save those into an analysis staging area. The difficult task is now for the user to design an analysis that combines the results from these analyses into a new model where data and analytics are used to create a new view of the extent of the experiment. In our example, we see that we have information on Griscelli's disease and we have some SNP and expression information on patients with and without Griscelli's disease. Given that the user was able to find an area on the chromosome with a high LOD score indicating that the SNPs from several case-control and pedigree experiments all point to one region of the chromosome containing a few hundred genes, and given that expression information exists between two cell lines, one with a normal MyoV gene and one with a mutant phenotype (Griscelli's disease phenocopy), we can now use the software to filter through all the expression data for genes within the chromosomal region found from the linkage analysis. The software may be sophisticated enough to identify a pattern of genes that show differential transcriptional response between normal and Griscelli's patients indicating some pleitropic transcriptional activity due to a mutant MyoV gene, a transcriptional feedback loop that unambiguously identifies novel MyoV-related pathways. If the analyst saw this same pattern in patients with Parkinson's disease, he might be able to identify a path of analysis that might not have been obvious before. Given that there are details or analyses that researchers normally ignore (e.g., phylogenetic trees that show relationships between genes), this analysis could illuminate candidate genes that might show interesting responses.

12.3.1. Maintenance, updates, and integration

Many specialized public databases are incontrovertibly invaluable to the biologist. Among these are LocusLink, Unigene, RefSeq, SwissProt, dbSNP, KEGG, BIND, and others. Much of today's genomic information relies on the most recent build of the NCBI reference human genome sequence. Build 34 (the most recent build so far) of the human genome is used by commercial and open-source software products in order to locate SNPs and genes at a precise location on each chromosome. Many readers will know the NCBI-supported databases, but other repositories have become indispensable as well: the BIND database of protein interactions (www.bind.ca), the KEGG pathway database at Kyoto, Japan (www.kegg.org), and the Brookhaven Protein Database (www.pdb.org). The PDB structural database has not grown as quickly as it has in the past but it contains a wealth of structural information about proteins, ligands, small molecules and conjugates. It is incumbent upon a biological federation of databases to update itself whenever possible by gathering the information it needs to bring some of this data in house. Most times a virtual link works just as well as bringing data in-house, but in the case of the human genome, many pieces of software need to know the physical locations of SNPs and other genomic features and thus the latest genome build is important. It is necessary, therefore, to occasionally download the latest version from NCBI to keep that information current. Keeping information updated is very straightforward because as data is updated in remote databases, a simple query to the specific database would return identical information about a particular query until such information is updated. After the update, the user community would know that information has changed and the old version is no longer the latest format.

Integration of this data is key to the practical use of this information in a way that assists the systems biologist. Integration consists of taking publicly available biological information and ensuring that it matches annotation-wise the information contained within the local database. Note that integration of data implies that custom or commercial software can access the data contained within these databases as easily as any well-versed bioinformatician.

Sharing of information has become a principle concern in these days of multicenter collaborative efforts. A large center should be able to share data and information readily with any other group that has Internet access. The growth of the Internet has ensured that most scientists have full access to a high-speed T1 line, and VPN and secure socket layer (SSL) have made encryption technology simple and painless for the end user. Data sharing is becoming more than simply identifying a valuable datafile in a public microarray database. Now users wish to share data among colleagues and collaborators in a way that increases the effectiveness of distance collaborations. Consortia of like-minded researchers (e.g., the NINDS/NIMH Microarray Consortium, http://arrayconsortium.tgen.org) are espousing the usefulness of providing data to members of the consortium while providing an easy-to-use web-based interface for uploading and downloading data, data analysis, and sharing project information and publications. These groups indicate a trend where focused groups are relying on the power of the Internet

to enhance collaborations. Data sharing becomes increasingly useful when the sponsoring organization (for expression array data, available at www.ebi.ac.uk/arrayexpress, www.ncbi.nlm.nih.gov/geo, and others) includes direct database access, advanced queries, and high-speed uploads and downloads. Security would be enhanced, multiple levels of data access would ensure project data can be shared at various levels—read only, read/write/and delete. Other types of sharing can exist in a proposed multitype federated biomedical database system: the user can query multiple or single types of data and request that data in the form of an XML file. This is one of the simplest methods for retrieving common data because it relies strictly on the native format of the database itself. Creating an XML file from data using specific field names and entity relationships directly from the DTD itself is very straightforward. The resulting data can be turned into a text representation of an XML file, which can then be zipped and sent via secure FTP. Sharing also depends on network security, and the type of security appropriate for biomedical data is more easily implemented in a federated system such as that seen in Figure 12.4. Network security can be structured by the IT department, but database security and permissions can be handled by the database administrator in conjunction with expert domain knowledge from scientists and analysts. Many security features are simply part of modern databases; these features have become invisible to the end user or the administrator, enhancing and simplifying data sharing and data availability.

12.4. Mathematical and computational tools for computational systems biology

Biological data gathered through various measurement technologies are subject to mathematical and computational tools for interrogation, analysis, and modeling.

12.4.1. Identification of disease subtypes and molecular markers

Understanding regulatory mechanisms that drive a biological system to a specific phenotype requires the ability to isolate the system regulated under a causative mechanism. The observations made from mixed pools of transcriptional systems each of which is governed by different mechanisms may prevent us from getting correct inferences and interpretations of the outcome of the analysis. This is one of the reasons that the experiments carefully designed to explore a specific biological context result in observations that are more easily understood and interpreted. However, we often do not have that much control over how to perform experiments or even how to collect the data. For example, many studies involving human subjects, that is, patients, or other live animals, are subject to certain protocols and we cannot arbitrarily design experiments that use excessive numbers of living subjects. Tumor samples or other disease-related tissues get even more complicated as issues of tissue acquisition and patient consent create complexities that are hindering such analyses. Therefore, it is critical to have computational or statistical methods to sift through this heterogeneous set of observations and sort them out to more homogeneous observations in biological contexts.

One of the most popular ways to identify this rather homogenous set of observations is clustering. By grouping sets of genes that share similar transcriptional expression patterns across multiple experiments (genewise clustering) or sets of experiments with similar expression pattern across multiple genes (experimentwise clustering), one can find sets of genes that might be coregulated under the same transcriptional regulation or a set of experiments that might have resulted from the same transcriptional regulation. There exist many different ways to define "similarity" between genes and experiments such as correlation, Euclidean distance, rank correlation, and so on. Also, there are various algorithms to find clusters (sets) of genes/experiments given such a similarity measure. Clustering has been quite useful in analyzing microarray data and identifying sets of coregulated genes [18, 21, 23, 74, 75, 76]. Such an approach is called unsupervised since there is no "teacher" that guides the process of learning to distinguish either sets of genes or sets of experiments based on common rules. While it sometimes reveals informative results, one can often find even more informative sets of genes if some prior biological knowledge or statistical occurrence is known (known as a prior in Bayesian statistics). Another interesting and very promising approach along this line is to utilize a gene clustering tool such as the GO browser (gene ontology) or other descriptive set of data that categorizes genes by function [77].

Once a different set of biological contexts are identified, the next step is to identify a set of genes or perhaps a single gene that might be a cause or element of a causative agent for a specific biological context. Statistical approaches such as Student t test, ANOVA, ANCOVA, MANCOVA, and others are typically used to score genes according to each gene's capability to discern its target context from others using discrete or continuous descriptors, resulting in a set of genes that can be used as molecular markers to separate, for example, disease subtypes. Various other approaches, statistical or algorithmic, have been tried with varying success. The use of the t test [78] or its many variants such as SAM [79] in microarray data is well accepted. They are single-gene-based approaches and rank a gene high if that gene has compactness within its class but significant separation between classes.

There are other single-gene-based approaches such as TNoM score [80], time-series analysis [81], PRIM [82] which uses false negative errors as score, false discovery rate (FDR) [83, 84], local-pooled-error (LPE) [85] and nonparametric approach [86]. Approaches that factor more than a single gene into the analysis include ANOVA [87], strong-feature set [88], gene shaving [89], Bayesian approach [90], and Genes@Work [91]. Most of these algorithms use some amount of statistics to evaluate each gene or gene list and each statistic is checked for significance.

For either diagnostic or prognostic purposes, molecular markers that are identified via feature selection methods described above can be used to construct classifiers. Many traditional learning algorithms have been used for microarray data. Linear discriminate analysis (LDA) [92, 93], k-nearest neighbor (k-NN) classifier [80, 93], decision tree (DT) [80, 93], support vector machine (SVM) [80, 93, 94], clustering-based algorithm [80], and kernel-based classifier [88, 95] are among

them. When classifiers are designed, classifier assessment, specifically its predictive power and sensitivity, is an issue well documented in the pattern classification community [96, 97, 98, 99, 100, 101]. There are various methods to evaluate estimated errors, mostly a function of the number of samples used in the design and the complexity of the classifier being designed. This issue has confounded the application of traditional rule-based and statistical-based learning approaches to data analysis due to the lack of sufficient numbers of samples to validate the significance of estimated errors. Recently, some efforts have been made to bring this issue to the attention of the community [93, 102, 103, 104], and some have tried to overcome the small sample issue [105, 106] or have compared the quality of error estimation methods [107, 108]. While there may be many more statistical issues to be resolved, at least a few studies have resulted in promising outcomes with strong associations to clinical and prognostic predictions [109, 110].

12.4.2. Machine learning for predictive functional relationships among genes

Identifying molecular markers for a certain type of disease helps physicians properly and efficiently diagnose the disease based on phenotypic or genotypic observations. However, it may not provide adequate insight to develop or implement an effective treatment plan to cure the disease unless the whys and hows of certain steps of molecular events occur to lead to the specific phenotype. Hence, learning about functional/causative relationships more effectively, at the molecular level, is very important in understanding and curing disease. Having molecular markers is analogous to having a list of parts for an automobile. It helps in the process of building a vehicle but it is not sufficient. We must understand how each part interacts with every other part to create functioning subassemblies that carry out certain processes for an entire functioning automobile. Micorarray data and other high-throughput measurement technologies present a unique opportunity to study the functional relationships among genes at a global scale. Systematic perturbation using technology such as gene silencing using antisense RNA [111, 112] or RNA-mediated interference [8, 66, 113] also dramatically increases our capability to identify those modules, or subassemblies, that work to bring a cell back to equilibrium.

The ultimate goal is to construct a comprehensive mathematical model, learn the possible parameters from observable data, and synthesize this model *in silico* for further analysis with extensive simulations and validation through refined observations. For example, a computational model can be used to predict the behavior of a system under certain conditions that may not be easily set up in a physical laboratory. While not strictly impossible with our current technology, we will require much more refined and detailed observational technologies, more data processing power, and new algorithms that can intelligently combine data types into a data stream that can be reconstructed to a level that allows us to build comprehensive mathematical and computational models for biological systems. Therefore, the effort to learn the predictive relationships found within the transcriptional

activity (currently the most applicable measurement technology for systems biology in terms of modeling biological networks) is still laudable at the very least and profoundly practical at best. For example, although it is feasible to perturb and monitor a cellular system *in trans* using antisense inhibitors of mRNA [111, 112] or RNA-mediated interference [8, 66, 113], it is not always practical to design and implement perturbation experiments in mammalian cells since they are not typically observable in a controled environment—the enormous amounts of confounding factors cannot be easily accommodated. It is therefore important to understand the extent to which networking between genes and other cellular components can be learned from steady-state observations. Although time course information is preferred, it is often impractical from a financial standpoint to hybridize enough arrays to obtain sufficient data. Another critical issue in learning a predictive relationship among genes is the lack of sufficient observations, that is, insufficient number of samples. There are various statistical issues; some have tried to bring attention to the issue [106, 114] with varying success. There have been many such efforts in the last five or six years.

Various mathematical and algorithmic approaches in conjunction with appropriate biological experimental data were used to alleviate this problem. General reverse engineering algorithm (REVEAL) with a Boolean network framework and the use of mutual information have been developed and applied to both simulated and real datasets [33]. Other REVEALs were also used for microarray data [115, 116], such as the ones based on singular-value decomposition [52] and genetic algorithm [115]. A reverse engineering approach with carefully designed biological experiments based on systematic perturbation seems to yield the highest degree of success for the reconstruction of the topology of genetic networks and has been shown in simulated data as well as *Drosophila melanogaster* datasets [67] and yeast [117]. Bayesian networks also showed their applicability to learning predictive relationships between genes based on microarray data [118]. Parametric estimation with a differential-equation-based model was applied to a nine-gene subnetwork of the SOS pathway in *Escherichia coli* to learn structure and function [30]. Similar to reverse engineering, rule-based learning has been tried on expression data in conjunction with knowledge information such as the GO [119, 120]. The result of these efforts is similar to the automobile assembly allegory above. When one knows the parts that make up the automobile, one knows the smallest components that are assembled to form functional modules, but one does not know the modules *per se*. When one knows some of the parts that contribute to a certain effect, one can narrow down that function to a small number of parts that contribute to that function and now we can assign some causation. When we wish to know only those functions that cause a completed automobile to veer violently to the left, we can look at a normal and problematic automobile and discover those parts that are faulty (e.g., a tie-rod end or steering damper or a combination of both). This is considered a search for a signature, or fingerprint—a small list of parts (genes) whose malfunction leads to a disease phenotype through direct or indirect causes. Occasionally researchers find some obvious downstream effects, perhaps cell wall permeability or decreased cell surface receptors that are

the direct cause of the disease phenotype. However, a closer look reveals that a single transcription factor with a mutation in the binding site has caused all of the downstream effects and is in fact the primary causative agent, much as a single stripped bolt can cause improper operation of an entire steering assembly in an automobile.

Not only gene-gene interactions, but also protein-protein interactions are necessary regulatory mechanisms to build insightful models, and they have been studied at great length to learn those relationships based on high-throughput data screens [121, 122]. Synthetic lethal screens and functional validation using combinatorial application of therapeutics are immensely useful for validating predictive models but the number of combinations and the computational load can be quite overwhelming and time consuming.

An important issue in learning functional or predictive relationships among genes or proteins is whether one can or cannot learn the relationships from steady-state observations, let alone causative relationships. It is a common belief in a nonlinear dynamic theory that one needs to sample dynamic systems at a high enough frequency, to learn about the system dynamics, and to begin to model the system. This is problematic in systems biology since most of the biological observations are designed for steady-state observations of cells, much less frequent enough to satisfy this condition. Nonetheless, many have tried to use steady-state observations to learn predictive relationships among genes and proteins, as described above, and often interesting relations among genes have been found, although none could claim to identify causative relationships until they are validated in wet-lab experiments. Recently, however, Bayesian or dynamic Bayesian frameworks have partially overcome this issue [34, 118, 123]. This approach seems promising in the sense that it does not explicitly require high-resolution sampling of time-series data. There also exist algorithmically improved methods to elicit causal relations based on their conditional dependencies [124, 125, 126].

12.4.3. Modeling gene regulatory networks

While new kinds of data are presently being used to identify molecular signatures with new diagnostic targets for diseases, in order to truly benefit from the volume and complexity of this data and understand the underlying processes that result in the markers, researchers also require computational and mathematical modeling of the biological activity under investigation. Understanding the biological mechanism underlying the cellular process may lead to significant advances in cell biology, drug development, and medicine. Therefore, it is increasingly clear that in order to enhance our knowledge about functional modules such as transcriptional regulation, it is vital to build mathematical and computational models for cellular processes with sufficient accuracy to make reasonable predictions about cellular mechanisms. Hence, synthesis of mathematical models for cellular processes, in silico simulation and analysis, and (preferred) biological validation will play fundamental roles in systems biology to understand roles and causality in cellular systems.

Modeling is a process by which a system is abstracted in order to simplify and modularize the relationships between functional elements of a system. A good model retains enough of the original system's characteristics—the flexibility, the dynamics, the response and accommodation to stress—to predict either a verifiable outcome within certain constrained parameters or an outcome that predicts a living response to a perturbation with input boundaries. As increasingly better measurement techniques evolve, we have better methods by which to test our system, and our predictive model becomes more robust. In addition, the modeling of a process and its subsequent analysis also drives intuition and insight to further develop new measurement and analysis techniques.

So far there has been, albeit with limited success, a significant amount of effort put into practice to construct a mathematical framework in order to model biological systems [28, 29, 53, 68, 127, 128, 129, 130, 131, 132]. The Boolean network model [26, 27, 44, 49, 133, 134, 135, 136, 137], Bayesian framework, differential-equation-based models, and linear models are among them. Random Boolean networks, first proposed by Kauffman [135, 138], were extensively studied to show that it possesses such biological properties as stability (attractor) and evolvability. Albert et al. have shown that the Boolean network, in opposition to the notion that Boolean networks can only qualitatively describe a genetic network, could also model the topology of regulatory interactions to predict the expression patterns of the segment polarity genes in *D. melanogaster* in a quantitative manner [139]. In mimicking incidental or intentional perturbation of gene regulation, Shmulevich et al. [140, 141] studied the stability and transition time of the model in response to a specific perturbation of transcriptional status. In another approach to mathematical modeling based on differential equations, Hasty et al. [68, 142], Ozbudak et al. [59], and Thattai et al. [58] studied intrinsic and extrinsic noise observed in cellular systems as well as dynamic evolution, and regulatory control of cellular systems.

There are a few computational tools available to help construct fairly complex gene regulatory networks, mostly based on differential-equation-based models; E-cell (http://www.e-cell.org) [143, 144] is a tool developed by a group of international collaborators to reconstruct biological phenomena in silico and simulate the system for the prediction of cellular behavior. genomic object net (available at http://www.genomicobject.net/) [145, 146] provides similar functionality.

Once such mathematical and computational models for cellular systems are available, it is paramount to simulate in silico systems to predict biological behavior under conditions that may be difficult to impose on a target system—this is one of the strongest reasons for developing a biological model. The conditions may include the one not possible in a natural environment but may have clinical value, such as a key mutation in a gene. Such prediction results could provide a new hypothesis that could be validated in the lab. Another aspect of the analysis of *in silico* models is to study system behavior in both transitional state and steady state (terminal, or endpoint) and to identify the properties that help maintain stability. Homeostasis, the ability of cells to maintain life by facilitating an internal

environment compatible with survival, and rapid transitions between metastable states, are among those cellular systemic properties.

The study of such behavior—robustness, stability, and topology—is getting the focus of scientific research these days. Robustness [10] is characterized by properties such as adaptation (the ability to cope with environmental changes), parameter insensitivity (a system's relative insensitivity to specific kinetic parameter values), and graceful degradation (the slow degradation of cellular functions after damage, rather than catastrophic failure). Mechanisms to maintain robustness include:

(1) control (negative feedback and feed-forward),
(2) redundancy (surrogates),
(3) structural stability (intrinsic mechanisms that are built to promote stability),
(4) modularity (subsystems are physically or functionally insulated so that failure in one module does not spread to other parts and lead to system-wide catastrophe).

Scale-free topology [147], ensemble [133, 148], canalizing functions [133], and postclasses [149] are among the many mechanisms that describe Boolean networks' capability to mimic such behavior of cellular systems. Recently, Kim *et al.* [150] used Boolean networks with steady-state observations to construct a finite-state Markov chain model whose transitions depend on state-dependant multivariate conditional probabilities between gene-expression levels. This model produced steady-state distributions closely approximating the observations made in microarray data and exhibiting only a limited number of states that possessed significant probability of occurrence. This behavior is congruent with biological behavior as cells appear to occupy only a negligible portion of the state space available to them.

12.5. Supercomputing and parallel applications

Many of the aforementioned analytical techniques are highly computing intensive, and require many months of computer time to be completed within a feasible time span. Several physical approaches exist within the supercomputer realm as follows.

(1) Cray, or vector-style supercomputers (shared-memory multiple vector processors) use several high-speed vector processors that access common memory running at the CPU core speed (typically many gigaflops) and rely on code specifically written for vector and floating point calculations.

(2) MPP (massively parallel processors) machines such as Paracel (available at www.paracel.com) and the now defunct TimeLogic systems utilize many thousands of inexpensive CPUs physically sharing memory on the same bus. These machines do not access memory at the core CPU speed.

(3) Beowulf-type machines (cluster processors) are discrete machines that share neither CPU bus nor a memory bus. A Beowulf cluster relies on high-speed external interconnections to transfer data to and from head nodes and between

slave nodes. Interconnections such as Myrinet or Gigabit Ethernet are significantly slower than shared physical memory.

(4) The distributed computing systems, such as SETI@Home, distribute fixed-size work units that are disseminated via the Internet to available clients.

Each of these computer hardware architectures is optimized for certain problems, but none works well for all problems in systems biology. Cluster-based supercomputers are well matched for many applications in clustering, classification, partitioning methods, and combinatorics heavy selection algorithms. Well-organized and flexible parallel MPI (message passing interface) libraries exist for C++, Java, perl, python, Matlab, and so forth and are ideal for parallelizing many of the problems that currently exist in bioinformatics and microarray analysis. Many of these algorithms follow the Monte Carlo scalability, that is, many problems do not need to share memory and are distributable in small self-contained packets, and the results can be easily combined after the process is completed. If a calculation needs to access the results from a previous calculation, we are unlikely to find an adequate or efficient method for increasing the parallel nature of the algorithm. Many statistical methods are highly serial, as the input to one calculation is the result from the previous calculation. Parallel statistical computing remains a difficult challenge to face. The parallel computer system in place at the Translational Genomics Research Institute and Arizona State University (TGen/ASU) has been used for many of the aforementioned problems. The system is an IBM xSeries cluster of 1024 Xeon 2.4 GHz processors (512 dual processor nodes) with 1M level 1 cache, 256 K level 2 cache, and 128 K level 3 burst cache. The CPU is a CISC processor and each node is an independent SMP machine with 2 CPUs, 2 G of RAM, and 60 G of hard drive space running the Red Hat Linux OS. Each node communicates with other nodes using Myrinet interconnections and Gigabit Ethernet connections to the head node. The system can accommodate both multiple program multiple data (MPMD) and single program multiple data (SPMD) environments (Hwang, 1993).

Classification and other iterative and heuristic algorithms are especially well-suited for parallelization. For example, the strong-classifier program performs an exhaustive search of combinatorial space across an immense feature set [88]. Typically, one would evenly distribute the computational load among all available processors, thereby reducing the load imbalance; however in practice this typically requires dynamic assignment and more overhead. Dividing the total number of *s*-classifiers across the total set of processors was inappropriate for less than 64-bit systems, so a sub-optimal distribution scheme was entailed. The most efficient method was to selectively modify each member of a "work group" using a predictive method for complexity. Each work unit was assigned a complexity value and sent to a CPU for processing. Load balancing was performed from the original calculation of complexity and task distribution by the head node.

Coefficient of determination (CoD) [35, 151] has typically been a challenge for parallelizing efficiently since uniform processor loading does not sufficiently accommodate the differential calculation speed of every combination of target genes necessary to identify those predictor genes that appear to exert influence on

the target genes. This algorithm is extremely time intensive due to the large number of combinations that need to be analyzed, and the final large sort of the data. Load balancing on parallel machines can be static or dynamic; in the case of CoD a dynamic load partition and assignment scheme was utilized based on partitioning the data and mapping computation and data modules onto separate processors. A weighting scheme was used to avoid lulls in processing time. Each clock cycle that was voided decreased the overall efficiency and it was noted that many algorithms that utilize time-independent but low-order combinatorial calculations benefit from dynamic CPU loading and monitoring. Often low-frequency monitoring to save clock cycles results in a high degree of random latency, and in the case of CoD it was found that a mid-level monitoring scheme was useful for maintaining full CPU load on each of the processors in a typical 32+ CPU Beowulf system.

12.6. An application: deciphering the Wnt5a signal cascade

We now present an exemplary study in which we used various mathematical and computational tools to decipher the WNT5a signaling cascade.

Over the past five years, the various researchers associated with this application have been developing a mixture of biological and mathematical observations, reagents, and approaches to make inferences about genomic regulatory networks [41, 88, 152, 153, 154, 155, 156, 157, 158, 159, 160]. While these have been applied to a variety of biological systems, all of the types of inference-generating methodology have been consistently applied to melanoma.

Figure 12.5 depicts a putative outline of signaling, given the current state of knowledge from one of the focal points of this study. The focus in this case is the network of elements of the Wnt5a signaling pathway that deals with the phenotypic change from a less motile, less aggressive cell to a much more motile, more invasive cell. The diagram contains information based on experimental observations, and the mathematical treatment of the data, as well as the integration of the models with what is known about the genes from genetic and biochemical characterizations in a variety of both mouse and human contexts.

12.6.1. Dataset: gene expression profile

The gene expression profiles used in this example were obtained from 31 melanoma samples and 587 genes [156]. In [156], total mRNA was isolated directly from melanoma biopsies; fluorescent cDNA from the message was prepared and hybridized to a microarray containing probes for 8150 cDNAs (representing 6971 unique genes). The data was also subject to stringent quality assessments to meet our quality standards.

12.6.2. Exploratory analysis: finding a subcontext of phenotype

Both quantitative and comparative measurements were applied for each gene. In this study, several analytical methods such as hierarchical clustering and multidimensional scaling (MDS) were performed to visualize the overall expression

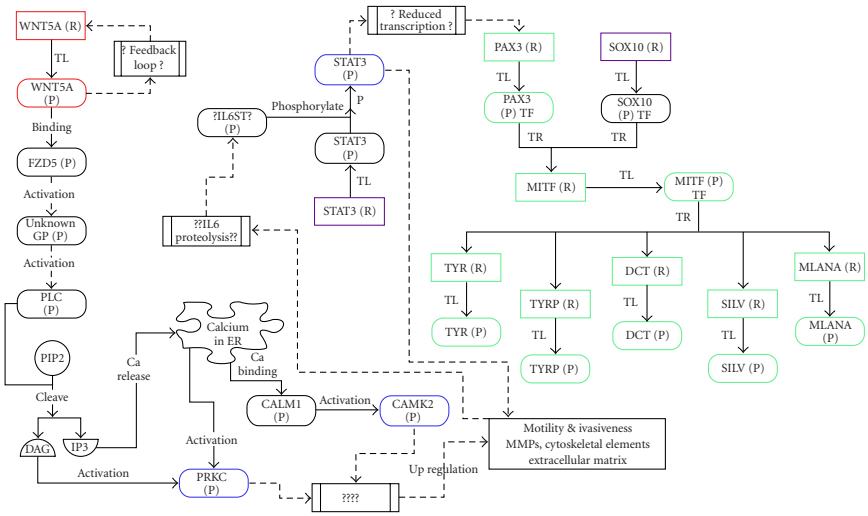


Figure 12.5. Features of the Wnt5a signaling pathway. A view of Wnt5a signaling in melanoma assembled from expression profiling, cell biology experimentation, mathematical analysis, modeling of experimental results, and incorporation of regulatory connections from external sources. The mRNAs are represented as rectangles, and proteins as ovals. Steps encompassing transcription (TR), translation (TL), and other biochemical processes are indicated as arrows with labels. Known changes in the relative abundance of species after Wnt5a stimulation are indicated by red and green (increased and decreased abundance). Phosphorylated protein species are indicated as blue. Speculative steps are indicated by dotted arrows, and speculative components are indicated with question marks.

pattern relationships among 31 cutaneous melanoma tumor samples. The clustering and MDS projected plots indicated that the 31 melanomas could be partitioned into two groups of 12 and 19 samples, respectively, as shown in Figure 12.6.

12.6.3. Discriminatory analysis: identifying regulatory components

In identifying genes that discriminate two different clusters in melanoma samples, a statistical measure was employed to generate a weighed gene list according to their impact on minimizing cluster volume and maximizing center-to-center intercluster distance. The 587 genes are the top 587 with highest weights out of the 6971 genes. Figure 12.7 shows the first few dozens of those discriminant genes. It is quite interesting to observe that gene expression pattern in the group with larger number of samples show much consistent pattern. In fact, this is congruent with the smaller size of the corresponding cluster visualized in the MDS plot in Figure 12.6b, the cluster inside the cylinder. This can be interpreted as follows, this set of genes is more tightly regulated by a cellular mechanism governing a specific cellular context that resulted in a specific phenotype observed for the set of samples; less motility. The same set of genes show less consistent molecular patterns outside the cluster, which may indicate that the same cellular mechanism is not operating

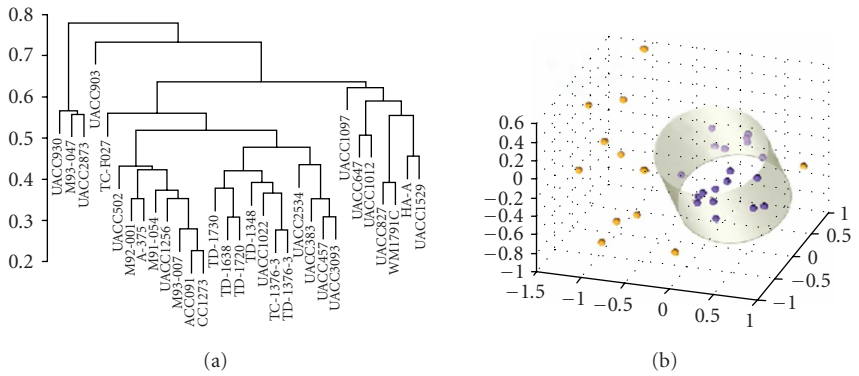


Figure 12.6. Hierarchical clustering and multidimensional scaling (MDS) plot reveal two distinct groups of samples.

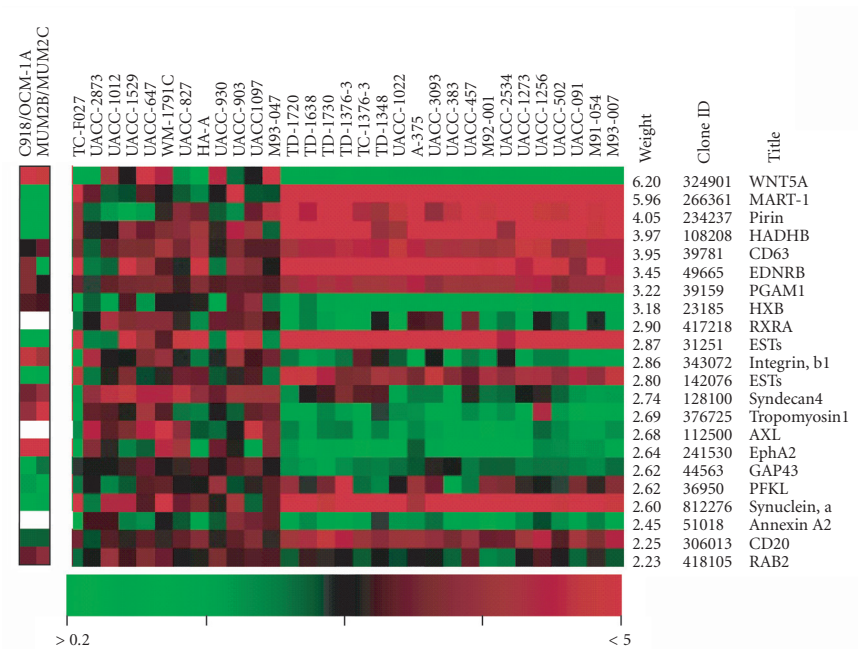


Figure 12.7. Gene expression profile with discriminant genes.

or at least not regulating the genes as tightly as it does within the cluster, hence, less controled behavior.

12.6.4. Prediction analysis: finding relationships and functions

Furthermore, each gene expression level was quantized to a ternary value that represents the abundance of mRNA produced by that gene in a particular melanoma

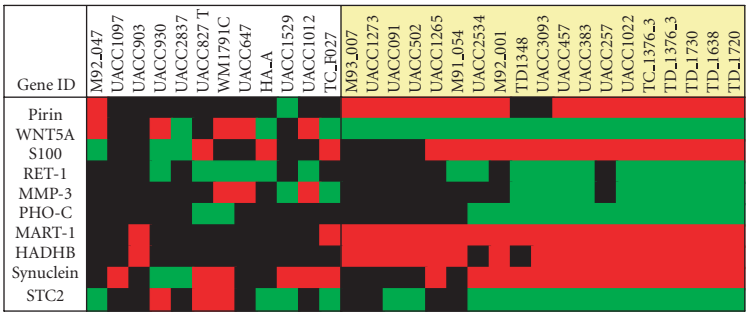


Figure 12.8. Quantized (ternary) gene expression data for 10 selected genes.

sample relative to the abundance of mRNA produced by that gene in a reference cell as shown in Figure 12.8. The values are overexpressed (red), no change (black), and underexpressed (green), relative to the reference.

The *coefficient of determination* (CoD) is a measure of the relative improvement in prediction accuracy owing to the presence of the observed variables, that is, how the combination of given genes (predictors) improves the prediction of the behavior of a target gene in comparison to the absence of predictors. It is mathematically defined as $\theta_{\text{opt}} = (\varepsilon_0 - \varepsilon_{\text{opt}})/\varepsilon_0$, where ε_{opt} is the error made by the optimal predictors and ε_0 is the prediction error in the absence of predictors. We first applied the CoD to determine multivariate intergenic relations, and relations between genes and external stresses arising from multiple conditions, such as radiation and chemical mutagens. In an effort to capture nonlinear decision making, we both considered a general logical approach [153] and then focused on additive (linear) regulation by considering perceptron-based prediction [154]. In both cases, we established the consistency of mathematical results with existing biological knowledge and speculated on the plausibility of strong multivariate relations in the data that did not correspond to previously known relations, some of which were validated later (unpublished). A key interest in applying CoD has been to determine the connectivity in genetic regulatory models [150, 159].

From the gene expression profiles used in the study, 50 genes capable of both predicting other genes as well as being predicted by other genes with high CoDs were chosen out of all genes. Then, 10 genes were further selected from 50 genes based on their roles in classifying malignant melanoma and known biological functionalities. Table 12.1 shows the best predictor combinations (genes) for each target and their corresponding predictive power (CoD).

This relationship is depicted in Figure 12.9 showing the wiring diagram of the selected 10 genes in which the genes are placed to reflect the influence between them; the higher the influence between genes, the closer they are placed. For example, Wnt5A and pirin are placed closest to each other, which is expected due to both their observed expression patterns and previously known relationship. However, the connection between WNT5A and MART-1 is of the greatest interest. It predicts that Wnt5a has a high influence on Mart-1.

Table 12.1. The 3-gene predictors for each target gene and their highest CoD values for 10 genes selected.

Predictor 1	Predictor 2	Predictor 3	Target	CoD
WNT5A (46)	HADHB (296)	ESTs (557)	pirin (6)	0.709
Pirin (6)	S100P (52)	RET-1 (63)	WNT5A (46)	0.683
WNT5A (46)	RET-1 (63)	synuclein (366)	S100P (52)	0.795
Pirin (6)	WNT5A (46)	S100P (52)	RET-1 (63)	0.625
S100P (52)	RET-1 (63)	HADHB (296)	MMP-3 (79)	0.700
MART-1 (147)	synuclein (366)	ESTs (557)	PHO-C (80)	0.920
Pirin (6)	WNT5A (46)	MMP-3 (79)	MART-1 (147)	0.793
Pirin (6)	WNT5A (46)	MMP-3 (79)	HADHB (296)	0.772
Pirin (6)	S100P (52)	MART-1 (147)	synuclein (366)	0.635
Pirin (6)	WNT5A (46)	PHO-C (80)	ESTs (557)	0.479

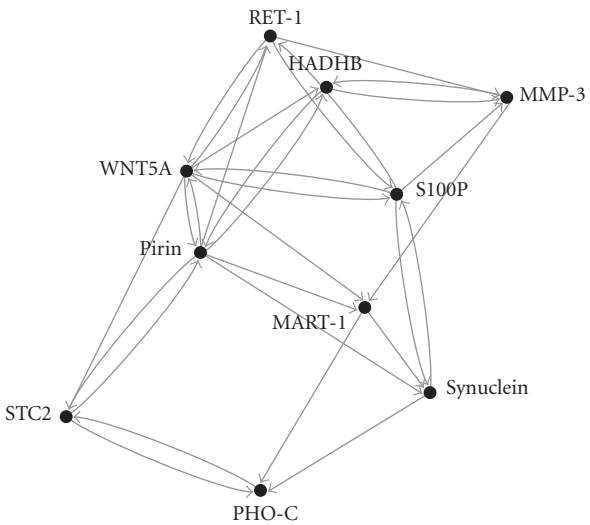


Figure 12.9. Predictive relationships among 10 genes of interest in melanoma.

12.6.5. Biological validation: follow-up

This proposed regulatory relationship has been followed up by detailed investigations of the relationship in other samples (Figure 12.10: Weeraratna and Nickoloff, unpublished), and the relationship is found to hold in most cases. Some of the details of regulation of a cascade that controls Mart-1 and a number of other genes expressed in melanocytes have been worked out in detail. These details are presented in Figure 10, starting with the induction of transcription of the MITF gene by the synergistic effect of the transcription factors, Pax3 and Sox10 on the MITF gene [161, 162, 163], and the subsequent induction of transcription of the TYR, TYRP, DCT, SILV, and MLANA (MART-1) genes by the Mitf transcription factor

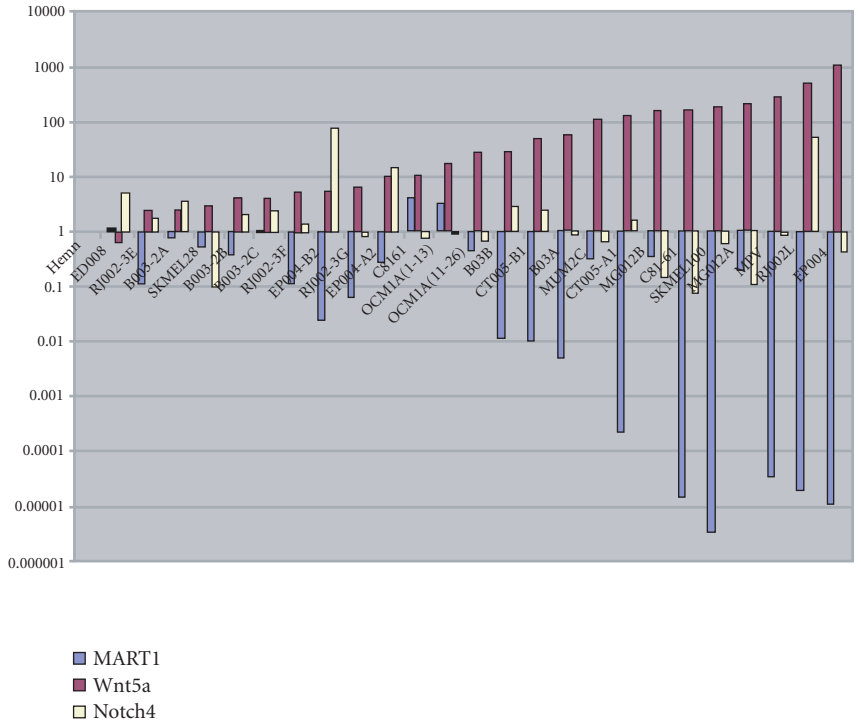


Figure 12.10. Anticorrelation of WNT5A and MART-1 mRNA abundance. The abundance of mRNA species for WNT5A, MART-1, and NOTCH4 determined by Q-RT PCR for a variety of melanoma cell lines and tumors relative to a melanoma cell line with low levels of each mRNA is shown. A high degree of anti-correlation is observed for WNT5A and MART-1, however NOTCH4, a protein frequently expressed in melanoma, which is not expected to be share regulatory information, has no apparent correlation to either WNT5A or MART-1.

[164, 165]. A great deal is known about this particular pathway since the loss of activity of some of the end products of the cascade can alter hair color in mouse and man and can be associated with a variety of human diseases such as albinism and impaired vision. Data consistent with downregulation of transcription of all members of this cascade starting with the PAX3 gene has been observed in our recent studies of very late stage melanoma tumors overexpressing the WNT5A gene (Bittner, unpublished). The ability to make such connections combining expression patterns, models based on inferences from these patterns, and existing knowledge of small segments of transcriptional networks suggests that these approaches will make it possible to use expression data to productively focus on possibly profitable experimentation.

12.6.6. Steady-state analysis: Markov chain simulation in the context of robustness

In the follow-up study of gene expression profiles of melanoma cell lines and the predictive relationships identified above, we constructed a finite-state Markov

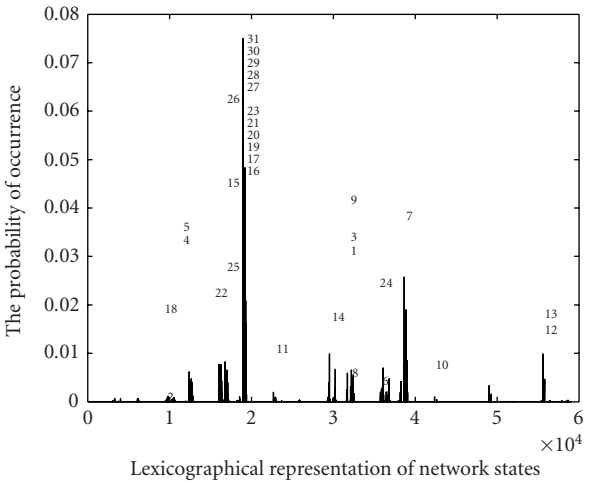


Figure 12.11. The steady-state distribution: each state is represented in a lexicographical order. Only a few states show significant probability at steady state. This predicts that the gene regulatory network abstracted in a probabilistic Boolean network has only a limited number of attractors at steady state, which is consistent with the fact that cells occupy only a negligible portion of the state space available to them.

chain model whose transitions depend on state-dependant multivariate conditional probabilities between gene-expression levels with the goal of determining whether the kinds of biological behavior observed and expected in biological systems could be captured in such a Markov chain model [150]. The Markov chain model contains n nodes, of which each node represents one of the N states composed of m genes selected. Each gene has a ternary value, which is assigned as overexpressed (1), no change (0), or underexpressed (-1). The state space of the Markov chain has 3^m states. For capturing the dynamics of the network, we consider a "wiring rule" such that the expression state of each gene at step $t+1$ is predicted by the expression levels of the other genes at step t in the same network. For each target gene, a set of predictor genes was chosen with the highest CoD value. Instead of using many possible Boolean functions that are independent of the state of the system, as in the PBN model [41], we use the state of predictor genes at step t and the corresponding conditional probabilities, which are estimated from observed data, to derive the state of target gene at step $t+1$. In the simulation, gene perturbation is also added to guarantee that the chain converges to a steady-state distribution [141, 150].

The steady-state distribution acquired from long-run simulation is shown in Figure 12.11. We concluded that the model produced steady-state distributions closely approximating the observations made in microarray data and exhibiting only a limited number of states possessing a significant probability of occurrence. This behavior is nicely congruent with biological behavior, as cells appear to occupy only a negligible portion of the state space available to them. The transition rules generated for the model produced localized stability. While the size of the

problem studied in this study is relatively small, it suggests that models incorporating rule-based transitions among states based on a Boolean network have a natural capability to mimic biology. The ability of such models to enhance our understanding of biological regulation should be further tested by systematically examining the characteristics of the rules and interconnections that lead to stabilization and switchlike transitions, and by building larger networks that incorporate more extensive prior knowledge of regulatory relationships and more extensive experimental observations of the different stable states the network can occupy to determine how accurately the model mimics these systems.

Figure 12.12 shows the marginal distributions of each gene, also at steady state. This distribution predicts that for example Wnt5a would be mostly downregulated at steady state while Mart-1 is upregulated at steady state. This is again consistent with the observations made through the gene expression microarrays shown in Figure 12.8, which is the measurement of cells at steady state.

12.6.7. Modular network identifications: growing from seed genes

By utilizing CoD and influence measure for probabilistic Boolean networks proposed by Shmulevich et al. [41], a method to grow a network given a smaller number of genes of interest, can be developed. The network is grown to be as self-contained and autonomous as possible, a key property of modularity. One such example when applied to melanoma gene expression profile is shown in Figure 12.13. The algorithm to construct this modular network is described in Hashimoto et al. [159]. This network was constructed from a pool of 587 genes and it consists of only 30 genes once it is constructed. This is a set of genes that maximize the overall predictability of this self-contained network. In the network diagram, the directed edges are not necessarily meant to indicate the causal relationships, but rather the influence of a gene on another in terms of information flow. However, we have shown an example in Figure 12.8 that even this crude level of abstraction of a gene regulatory network can lead to further development of new knowledge with the help of biologists' insight and biological experiments.

Key information needed for the network-growing method is a set of small number of genes of interest: *seed genes*. How to identify the seed genes of interest is very critical when applying this algorithm. We can use prior biological knowledge to choose seeds of interest. However, when such prior knowledge is not available, which is quite often the case in biology, we need a way to systematically identify those seed genes. Since the method is also computationally intensive and its computational complexity grows exponentially as the number of genes grows, methods to rapidly screen out irrelevant genes will be practically very useful. A typical method to fit this need is clustering which groups genes according to their expression patterns hoping that such genes with similar gene expression patterns belong to similar biological processes representing a self-contained module. Although they are quite useful methods and oft used in genomic data analysis, they do not extend beyond the identification of pairwise linear relationships among genes. Segal et al. [77] proposed an interesting method to identify regulatory modules based on gene expression data and some biological prior knowledge.

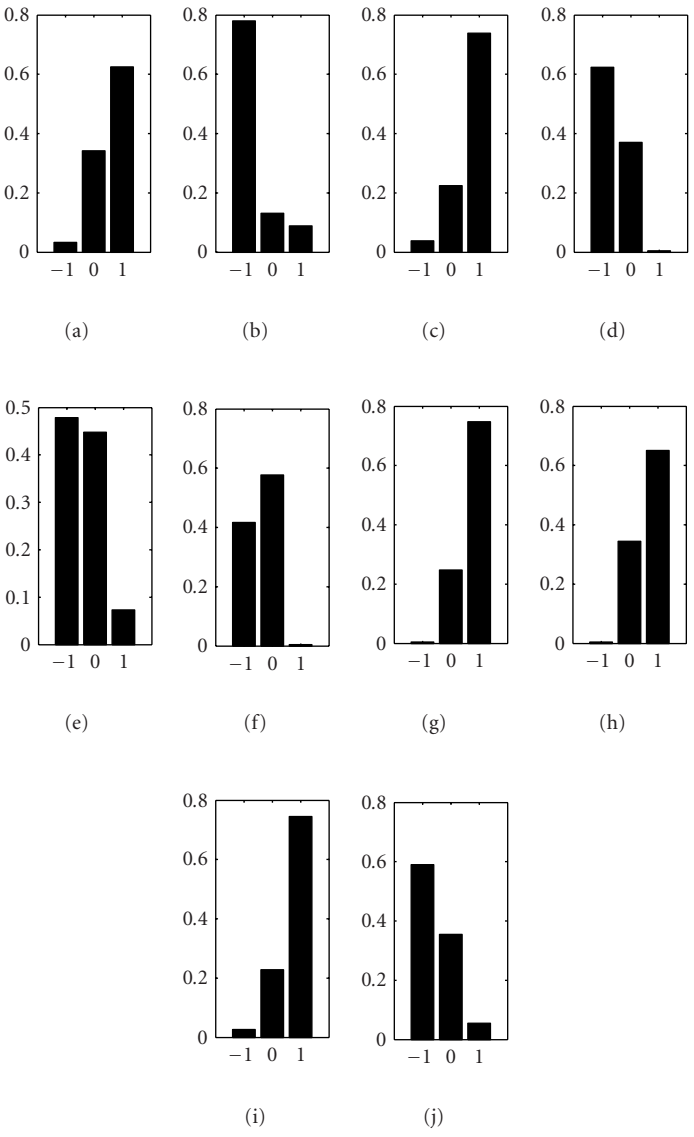


Figure 12.12. Marginal distribution of each gene at steady state: (a) pirin, (b) WNT5A, (c) S100P, (d) RET-1, (e) MMP-3, (f) PHO-C, (g) MART-1, (h) HADHB, (i) synuclein, (j) STC-2.

12.7. Conclusion

Computational systems biology is the set of methods and tasks that help us study computational aspects of systems biology, helping biomedical scientists know how

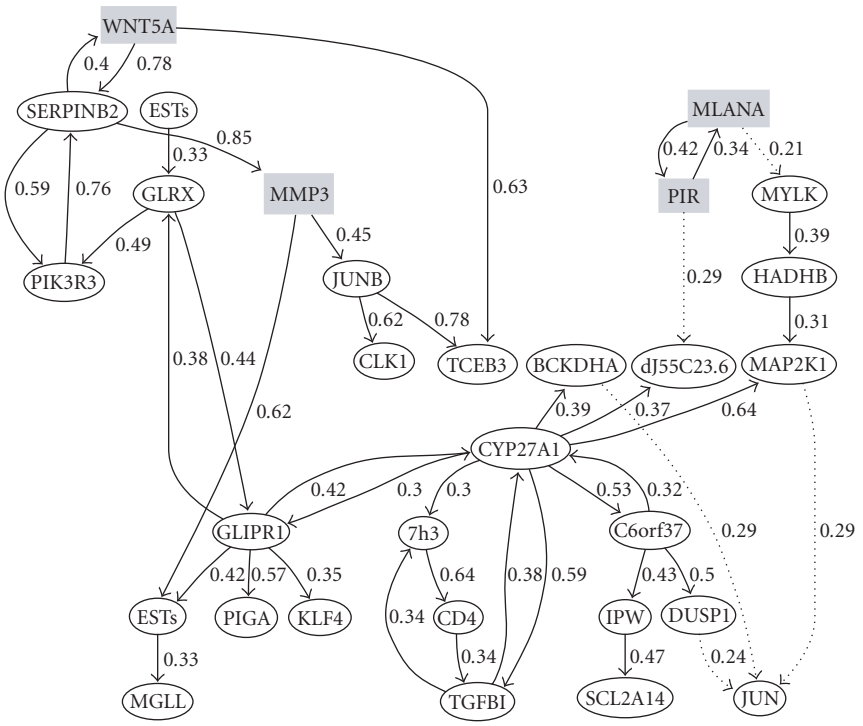


Figure 12.13. A modular network grown from a set of 4 genes; WNT5A, MMP3, PIR, and MLANA. This network is modular and self-contained in the sense that the set of genes are found to maximize the predictability between genes within the network without the help from genes outside the network. In other words, no other gene outside the network can be added without decreasing the overall predictability of the network.

cells operate as a system by developing various computational and mathematical tools that explain the function of independent modules, and then how these modules interrelate. This requires the development of tools at various levels of abstraction and with many different goals. When one wishes to find sub-contexts of cellular behavior, clustering and other exploratory tools are of interest. Discriminatory analysis would help identify a set of genes that are strongly associated with a specific cellular context such as a certain disease, therefore, becoming potential diagnostic markers. Simple statistical tools provide the basic understanding of how populations of observations represent the actual behavior of all cells. The problem of learning how genes are interacting with one another is often handled by prediction models and machine learning but too many times this fails to capture that mechanistic level of interaction among genes. The problem of modeling entire cellular systems is even more abstract and deals with a much higher level of understanding of the global interaction between genes. We are still at the early stage of learning and the development of the many necessary computational tools that are needed is proceeding, albeit at a slow rate. We are attempting for the first time

to successfully implement highly robust data storage and integration tools so that new algorithms and software products can have simplified access to many levels and types of data simultaneously and can integrate this data producing a model that is much more robust and accurate than one produced with a single type of data. One of the fastest growing fields in biology is the development of comprehensive and robust lexical standards that describe biological observations and phenomena. Biologists are working with database experts to precisely define the universe of data from a particular type of biological observation (e.g., the MAGE consortium scientists designed the MAGE object model through cooperative interaction with the OMG group). Once a firm set of international standards for text descriptions is available, the data stored in public and private databases will be much more amenable to mining, categorizing, and ultimately contributing to our understanding of the biological process. The systemic support for multitype data integration is also critical in systems biology to expedite the synthesis-analysis-feedback process and, in so doing, accelerate fundamental knowledge discovery about cells, diseases, and the delicate balancing act of life itself.

Bibliography

[1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[2] M. Chee, R. Yang, E. Hubbell, et al., "Accessing genetic information with high-density DNA arrays," *Science*, vol. 274, no. 5287, pp. 610–614, 1996.

[3] D. G. Wang, J. B. Fan, C. J. Siao, et al., "Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome," *Science*, vol. 280, no. 5366, pp. 1077–1082, 1998.

[4] C. M. Chen, H. L. Chen, T. H. Hsiau, et al., "Methylation target array for rapid analysis of CpG island hypermethylation in multiple tissue genomes," *Am. J. Pathol.*, vol. 163, no. 1, pp. 37–45, 2003.

[5] G. MacBeath and S. L. Schreiber, "Printing proteins as microarrays for high-throughput function determination," *Science*, vol. 289, no. 5485, pp. 1760–1763, 2000.

[6] H. Zhu, M. Bilgin, R. Bangham, et al., "Global analysis of protein activities using proteome chips," *Science*, vol. 293, no. 5537, pp. 2101–2105, 2001.

[7] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello, "Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*," *Nature*, vol. 391, no. 6669, pp. 806–811, 1998.

[8] N. J. Caplen, S. Parrish, F. Imani, A. Fire, and R. A. Morgan, "Specific inhibition of gene expression by small double-stranded RNAs in invertebrate and vertebrate systems," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 17, pp. 9742–9747, 2001.

[9] A. G. Fraser, R. S. Kamath, P. Zipperlen, M. Martinez-Campos, M. Sohrmann, and J. Ahringer, "Functional genomic analysis of *C. elegans* chromosome I by systematic RNA interference," *Nature*, vol. 408, no. 6810, pp. 325–330, 2000.

[10] H. Kitano, "Systems biology: a brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.

[11] M. Ehrenberg, J. Elf, E. Aurell, R. Sandberg, and J. Tegner, "Systems biology is taking off," *Genome Res.*, vol. 13, no. 11, pp. 2377–2380, 2003.

[12] T. Ideker, T. Galitski, and L. Hood, "A new approach to decoding life: systems biology," *Annu. Rev. Genomics Hum. Genet.*, vol. 2, pp. 343–372, 2001.

[13] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, "Stochastic gene expression in a single cell," *Science*, vol. 297, no. 5584, pp. 1183–1186, 2002.

[14] P. S. Swain, M. B. Elowitz, and E. D. Siggia, "Intrinsic and extrinsic contributions to stochasticity in gene expression," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 20, pp. 12795–12800, 2002.

[15] S. Solinas-Toldo, S. Lampel, S. Stilgenbauer, et al., "Matrix-based comparative genomic hybridization: biochips to screen for genomic imbalances," *Genes Chromosomes Cancer*, vol. 20, no. 4, pp. 399–407, 1997.

[16] D. Pinkel, R. Segraves, D. Sudar, et al., "High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays," *Nat. Genet.*, vol. 20, no. 2, pp. 207–211, 1998.

[17] H. Kitano, "Computational systems biology," *Nature*, vol. 420, no. 6912, pp. 206–210, 2002.

[18] U. Alon, N. Barkai, D. A. Notterman, et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 12, pp. 6745–6750, 1999.

[19] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *J. Comput. Biol.*, vol. 6, no. 3-4, pp. 281–297, 1999.

[20] J. DeRisi, L. Penland, P. O. Brown, et al., "Use of a cDNA microarray to analyse gene expression patterns in human cancer," *Nat. Genet.*, vol. 14, no. 4, pp. 457–460, 1996.

[21] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, 1998.

[22] C. Sabatti, L. Rohlin, M. K. Oh, and J. C. Liao, "Co-expression pattern from DNA microarray experiments as a tool for operon prediction," *Nucleic Acids Res.*, vol. 30, no. 13, pp. 2886–2893, 2002.

[23] P. Tamayo, D. Slonim, J. Mesirov, et al., "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 6, pp. 2907–2912, 1999.

[24] J. Qian, J. Lin, N. M. Luscombe, H. Yu, and M. Gerstein, "Prediction of regulatory networks: genome-wide identification of transcription factor targets from gene expression data," *Bioinformatics*, vol. 19, no. 15, pp. 1917–1926, 2003.

[25] Z. Bar-Joseph, G. K. Gerber, T. I. Lee, et al., "Computational discovery of gene modules and regulatory networks," *Nat. Biotechnol.*, vol. 21, no. 11, pp. 1337–1342, 2003.

[26] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," *Pac. Symp. Biocomput.*, vol. 4, pp. 17–28, 1999.

[27] T. Akutsu, S. Miyano, and S. Kuhara, "Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 331–343, 2000.

[28] T. Akutsu, S. Miyano, and S. Kuhara, "Algorithms for inferring qualitative models of biological networks," *Pac. Symp. Biocomput.*, vol. 5, pp. 293–304, 2000.

[29] H. Bolouri and E. H. Davidson, "Modeling transcriptional regulatory networks," *Bioessays*, vol. 24, no. 12, pp. 1118–1129, 2002.

[30] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp. 102–105, 2003.

[31] R. F. Hashimoto, S. Kim, I. Shmulevich, W. Zhang, M. L. Bittner, and E. R. Dougherty, "Growing genetic regulatory networks from seed genes," *Bioinformatics*, vol. 20, no. 8, pp. 1241–1247, 2004.

[32] M. Kaern, W. J. Blake, and J. J. Collins, "The engineering of gene regulatory networks," *Annu. Rev. Biomed. Eng.*, vol. 5, pp. 179–206, 2003.

[33] S. Liang, S. Fuhrman, and R. Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," *Pac. Symp. Biocomput.*, vol. 3, pp. 18–29, 1998.

[34] D. Pe'er, A. Regev, G. Elidan, and N. Friedman, "Inferring subnetworks from perturbed expression profiles," *Bioinformatics*, vol. 17, suppl 1, pp. S215–S224, 2001.

[35] S. Kim, E. R. Dougherty, Y. Chen, et al., "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.

[36] S. Aburatani, K. Tashiro, C. J. Savoie, et al., "Discovery of novel transcription control relationships with gene regulatory networks generated from multiple-disruption full genome expression libraries," *DNA Res.*, vol. 10, no. 1, pp. 1–8, 2003.
[37] A. Becskei and L. Serrano, "Engineering stability in gene networks by autoregulation," *Nature*, vol. 405, no. 6786, pp. 590–593, 2000.
[38] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: *in numero* molecular biology," *Nat. Rev. Genet.*, vol. 2, no. 4, pp. 268–279, 2001.
[39] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and S-system," *Bioinformatics*, vol. 19, no. 5, pp. 643–650, 2003.
[40] H. H. McAdams and L. Shapiro, "Circuit simulation of genetic networks," *Science*, vol. 269, no. 5224, pp. 650–656, 1995.
[41] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.
[42] K. Sachs, D. Gifford, T. Jaakkola, P. Sorger, and D. A. Lauffenburger, "Bayesian network approach to cell signaling pathway modeling," *Sci. STKE*, vol. 2002, no. 148, pp. PE38, 2002.
[43] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alché Buc, "Gene networks inference using dynamic Bayesian networks," *Bioinformatics*, vol. 19, suppl 2, pp. II138–II148, 2003.
[44] A. Silvescu and V. Honavar, "Temporal Boolean network models of genetic networks and their inference from gene expression time series," *Complex Systems*, vol. 13, no. 1, pp. 61–78, 2001.
[45] P. Smolen, D. A. Baxter, and J. H. Byrne, "Modeling transcriptional control in gene networks—methods, recent results, and future directions," *Bull. Math. Biol.*, vol. 62, no. 2, pp. 247–292, 2000.
[46] P. Smolen, D. A. Baxter, and J. H. Byrne, "Mathematical modeling of gene networks," *Neuron*, vol. 26, no. 3, pp. 567–580, 2000.
[47] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nat. Genet.*, vol. 22, no. 3, pp. 281–285, 1999.
[48] N. Barkai and S. Leibler, "Robustness in simple biochemical networks," *Nature*, vol. 387, no. 6636, pp. 913–917, 1997.
[49] S. A. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, no. 215, pp. 177–178, 1969.
[50] A. M. Sengupta, M. Djordjevic, and B. I. Shraiman, "Specificity and robustness in transcription control networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 4, pp. 2072–2077, 2002.
[51] G. von Dassow, E. Meir, E. M. Munro, and G. M. Odell, "The segment polarity network is a robust developmental module," *Nature*, vol. 406, no. 6792, pp. 188–192, 2000.
[52] M. K. Yeung, J. Tegner, and J. J. Collins, "Reverse engineering gene networks using singular value decomposition and robust regression," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 9, pp. 6163–6168, 2002.
[53] J. Goutsias and S. Kim, "A nonlinear discrete dynamical model for transcriptional regulation: construction and properties," *Biophys. J.*, vol. 86, no. 4, pp. 1922–1945, 2004.
[54] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.
[55] J. J. Fox and C. C. Hill, "From topology to dynamics in biochemical networks," *Chaos*, vol. 11, no. 4, pp. 809–815, 2001.
[56] A. V. Lukashin, M. E. Lukashev, and R. Fuchs, "Topology of gene expression networks as revealed by data mining and modeling," *Bioinformatics*, vol. 19, no. 15, pp. 1909–1916, 2003.
[57] J. W. Little, D. P. Shepley, and D. W. Wert, "Robustness of a gene regulatory circuit," *EMBO J.*, vol. 18, no. 15, pp. 4299–4307, 1999.
[58] M. Thattai and A. van Oudenaarden, "Intrinsic noise in gene regulatory networks," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 15, pp. 8614–8619, 2001.
[59] E. M. Ozbudak, M. Thattai, I. Kurtser, A. D. Grossman, and A. van Oudenaarden, "Regulation of noise in the expression of a single gene," *Nat. Genet.*, vol. 31, no. 1, pp. 69–73, 2002.

[60] W. C. Hahn and R. A. Weinberg, "Modelling the molecular circuitry of cancer," *Nat. Rev. Cancer*, vol. 2, no. 5, pp. 331–341, 2002.
[61] J. Coutinho, "RUNX1: transcription factor scores a hat-trick of autoimmune diseases," *Clin. Genet.*, vol. 65, no. 3, pp. 180–182, 2004.
[62] T. Dork, R. Fislage, T. Neumann, B. Wulf, and B. Tummler, "Exon 9 of the CFTR gene: splice site haplotypes and cystic fibrosis mutations," *Hum. Genet.*, vol. 93, no. 1, pp. 67–73, 1994.
[63] C. L. Lorson, E. Hahnen, E. J. Androphy, and B. Wirth, "A single nucleotide in the SMN gene regulates splicing and is responsible for spinal muscular atrophy," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 11, pp. 6307–6311, 1999.
[64] J. H. Kang, M. L. Hong, D. W. Kim, et al., "Genomic organization, tissue distribution and deletion mutation of human pyridoxine 5′-phosphate oxidase," *Eur. J. Biochem.*, vol. 271, no. 12, pp. 2452–2461, 2004.
[65] K. W. Kinzler and B. Vogelstein, *Colorectal Cancer*, McGraw-Hill, New York, NY, USA, 2001.
[66] S. Mousses, N. J. Caplen, R. Cornelison, et al., "RNAi microarray analysis in cultured mammalian cells," *Genome Res.*, vol. 13, no. 10, pp. 2341–2347, 2003.
[67] J. Tegner, M. K. Yeung, J. Hasty, and J. J. Collins, "Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 10, pp. 5944–5949, 2003.
[68] J. Hasty, D. McMillen, and J. J. Collins, "Engineered gene circuits," *Nature*, vol. 420, no. 6912, pp. 224–230, 2002.
[69] W. J. Blake, M. KAErn, C. R. Cantor, and J. J. Collins, "Noise in eukaryotic gene expression," *Nature*, vol. 422, no. 6932, pp. 633–637, 2003.
[70] F. J. Isaacs, J. Hasty, C. R. Cantor, and J. J. Collins, "Prediction and measurement of an autoregulatory genetic module," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 13, pp. 7714–7719, 2003.
[71] I. Biran, D. M. Rissin, E. Z. Ron, and D. R. Walt, "Optical imaging fiber-based live bacterial cell array biosensor," *Anal. Biochem.*, vol. 315, no. 1, pp. 106–113, 2003.
[72] I. Biran and D. R. Walt, "Optical imaging fiber-based single live cell arrays: a high-density cell assay platform," *Anal. Chem.*, vol. 74, no. 13, pp. 3046–3054, 2002.
[73] J. Ueberfeld and D. R. Walt, "Reversible ratiometric probe for quantitative DNA measurements," *Anal. Chem.*, vol. 76, no. 4, pp. 947–952, 2004.
[74] M. P. Brown, W. N. Grundy, D. Lin, et al., "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 1, pp. 262–267, 2000.
[75] A. J. Butte, L. Bao, B. Y. Reis, T. W. Watkins, and I. S. Kohane, "Comparing the similarity of time-series gene expression using signal processing metrics," *J. Biomed. Inform.*, vol. 34, no. 6, pp. 396–405, 2001.
[76] G. Getz, H. Gal, I. Kela, D. A. Notterman, and E. Domany, "Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data," *Bioinformatics*, vol. 19, no. 9, pp. 1079–1089, 2003.
[77] E. Segal, M. Shapira, A. Regev, et al., "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," *Nat. Genet.*, vol. 34, no. 2, pp. 166–176, 2003.
[78] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
[79] V. G. Tusher, R. Tibshirani, and G. Chu, "Significance analysis of microarrays applied to the ionizing radiation response," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 9, pp. 5116–5121, 2001.
[80] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 559–583, 2000.
[81] Z. Bar-Joseph, G. Gerber, I. Simon, D. K. Gifford, and T. S. Jaakkola, "Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 18, pp. 10146–10151, 2003.
[82] S. W. Cole, Z. Galic, and J. A. Zack, "Controlling false-negative errors in microarray differential expression analysis: a PRIM approach," *Bioinformatics*, vol. 19, no. 14, pp. 1808–1816, 2003.

[83] J. D. Storey and R. Tibshirani, "Statistical methods for identifying differentially expressed genes in DNA microarrays," *Methods Mol. Biol.*, vol. 224, pp. 149–157, 2003.
[84] J. D. Storey and R. Tibshirani, "Statistical significance for genomewide studies," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 16, pp. 9440–9445, 2003.
[85] N. Jain, J. Thatte, T. Braciale, K. Ley, M. O'Connell, and J. K. Lee, "Local-pooled-error test for identifying differentially expressed genes with a small number of replicated microarrays," *Bioinformatics*, vol. 19, no. 15, pp. 1945–1951, 2003.
[86] Y. Zhao and W. Pan, "Modified nonparametric approaches to detecting differentially expressed genes in replicated microarray experiments," *Bioinformatics*, vol. 19, no. 9, pp. 1046–1054, 2003.
[87] M. K. Kerr, M. Martin, and G. A. Churchill, "Analysis of variance for gene expression microarray data," *J. Comput. Biol.*, vol. 7, no. 6, pp. 819–837, 2000.
[88] S. Kim, E. R. Dougherty, J. Barrera, Y. Chen, M. L. Bittner, and J. M. Trent, "Strong feature sets from small samples," *J. Comput. Biol.*, vol. 9, no. 1, pp. 127–146, 2002.
[89] T. Hastie, R. Tibshirani, M. B. Eisen, et al., "'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns," *Genome Biol.*, vol. 1, no. 2, pp. RESEARCH0003, 2000.
[90] K. E. Lee, N. Sha, E. R. Dougherty, M. Vannucci, and B. K. Mallick, "Gene selection: a Bayesian variable selection approach," *Bioinformatics*, vol. 19, no. 1, pp. 90–97, 2003.
[91] A. Califano, G. Stolovitzky, and Y. Tu, "Analysis of gene expression microarrays for phenotype classification," *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, vol. 8, pp. 75–85, 2000.
[92] J. H. Cho, D. Lee, J. H. Park, K. Kim, and I. B. Lee, "Optimal approach for classification of acute leukemia subtypes based on gene expression data," *Biotechnol. Prog.*, vol. 18, no. 4, pp. 847–854, 2002.
[93] B. Wu, T. Abbott, D. Fishman, et al., "Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data," *Bioinformatics*, vol. 19, no. 13, pp. 1636–1643, 2003.
[94] Y. Lee and C. K. Lee, "Classification of multiple cancer types by multicategory support vector machines using gene expression data," *Bioinformatics*, vol. 19, no. 9, pp. 1132–1139, 2003.
[95] A. M. Bagirov, B. Ferguson, S. Ivkovic, G. Saunders, and J. Yearwood, "New algorithms for multi-class cancer diagnosis using tumor gene expression signatures," *Bioinformatics*, vol. 19, no. 14, pp. 1800–1807, 2003.
[96] M. Hills, "Allocation rules and their error rates," *J. Roy. Statist. Soc. Ser. B*, vol. 28, pp. 1–31, 1966.
[97] S. Raudys, "How good are support vector machines?," *Neural Netw.*, vol. 13, no. 1, pp. 17–19, 2000.
[98] S. M. Snapinn and J. D. Knoke, "An evaluation of smoothed classification error-rate estimators," *Technometrics*, vol. 27, no. 2, pp. 199–206, 1985.
[99] G. E. Tutz, "Smoothed additive estimators for non-error rates in multiple discriminant analysis," *Pattern Recognition*, vol. 18, no. 2, pp. 151–159, 1985.
[100] V. Vapnik and O. Chapelle, "Bounds on error expectation for support vector machines," *Neural Comput.*, vol. 12, no. 9, pp. 2013–2036, 2000.
[101] S. N. Attoor and E. R. Dougherty, "Classifier performance as a function of distributional complexity," *Pattern Recognition*, vol. 37, no. 8, pp. 1641–1651, 2004.
[102] R. L. Somorjai, B. Dolenko, and R. Baumgartner, "Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions," *Bioinformatics*, vol. 19, no. 12, pp. 1484–1491, 2003.
[103] E. R. Dougherty and S. N. Attoor, "Design issues and comparison of methods for microarray-based classification," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., pp. 93–112, Kluwer Academic Publishers, New York, NY, USA, 2002.
[104] E. R. Dougherty, "Small sample issues for microarray-based classification," *Comparative and Functional Genomics*, vol. 2, pp. 28–34, 2001.
[105] U. M. Braga-Neto and E. R. Dougherty, "Bolstered error estimation," *Pattern Recognition*, vol. 37, no. 6, pp. 1267–1281, 2004.
[106] M. Brun, D. L. Sabbagh, S. Kim, and E. R. Dougherty, "Corrected small-sample estimation of the Bayes error," *Bioinformatics*, vol. 19, no. 8, pp. 944–951, 2003.

[107] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?," *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.
[108] U. M. Braga-Neto, R. Hashimoto, E. R. Dougherty, D. V. Nguyen, and R. J. Carroll, "Is cross-validation better than resubstitution for ranking genes?," *Bioinformatics*, vol. 20, no. 2, pp. 253–258, 2004.
[109] L. J. van't Veer, H. Dai, M. J. van de Vijver, et al., "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.
[110] M. J. van de Vijver, Y. D. He, L. J. van't Veer, et al., "A gene-expression signature as a predictor of survival in breast cancer," *N. Engl. J. Med.*, vol. 347, no. 25, pp. 1999–2009, 2002.
[111] A. Nasevicius and S. C. Ekker, "Effective targeted gene "knockdown" in zebrafish," *Nat Genet.*, vol. 26, no. 2, pp. 216–220, 2000.
[112] M. Faria, D. G. Spiller, C. Dubertret, et al., "Phosphoramidate oligonucleotides as potent antisense molecules in cells and in vivo," *Nat. Biotechnol.*, vol. 19, no. 1, pp. 40–44, 2001.
[113] B. L. Bass, "Double-stranded RNA as a template for gene silencing," *Cell*, vol. 101, no. 3, pp. 235–238, 2000.
[114] D. Husmeier, "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks," *Bioinformatics*, vol. 19, no. 17, pp. 2271–2282, 2003.
[115] D. Repsilber, H. Liljenstrom, and S. G. Andersson, "Reverse engineering of regulatory networks: simulation studies on a genetic algorithm approach for ranking hypotheses," *Biosystems*, vol. 66, no. 1-2, pp. 31–41, 2002.
[116] M. Wahde and J. Hertz, "Coarse-grained reverse engineering of genetic regulatory networks," *Biosystems*, vol. 55, no. 1-3, pp. 129–136, 2000.
[117] N. Guelzim, S. Bottani, P. Bourgine, and F. Kepes, "Topological and causal structure of the yeast transcriptional regulatory network," *Nat. Genet.*, vol. 31, no. 1, pp. 60–63, 2002.
[118] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7, no. 3-4, pp. 601–620, 2000.
[119] T. R. Hvidsten, A. Laegreid, and J. Komorowski, "Learning rule-based models of biological process from gene expression time profiles using gene ontology," *Bioinformatics*, vol. 19, no. 9, pp. 1116–1123, 2003.
[120] J. S. Bader, "Greedily building protein networks with confidence," *Bioinformatics*, vol. 19, no. 15, pp. 1869–1874, 2003.
[121] S. M. Gomez, W. S. Noble, and A. Rzhetsky, "Learning to predict protein-protein interactions from protein sequences," *Bioinformatics*, vol. 19, no. 15, pp. 1875–1881, 2003.
[122] S. Tornow and H. W. Mewes, "Functional modules by relating protein interaction networks and gene expression," *Nucleic Acids Res.*, vol. 31, no. 21, pp. 6283–6289, 2003.
[123] C. Yoo and G. F. Cooper, "Discovery of gene-regulation pathways using local causal search," *Proc. AMIA Symp.*, pp. 914–918, 2002.
[124] J. Pearl, *Causality : Models, Reasoning, and Inference*, Cambridge University Press, Cambridge, UK, 2000.
[125] C. Glymour, "Learning, prediction and causal Bayes nets," *Trends Cogn. Sci.*, vol. 7, no. 1, pp. 43–48, 2003.
[126] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2000.
[127] H. Bolouri and E. H. Davidson, "Modeling DNA sequence-based cis-regulatory gene networks," *Dev. Biol.*, vol. 246, no. 1, pp. 2–13, 2002.
[128] M. A. Gibson and E. Mjolsness, "Modeling the activity of single genes," in *Computational Modeling of Genetic and Biochemical Networks*, J. M. Bower and H. Bolouri, Eds., pp. 3–48, MIT Press, Cambridge, Mass, USA, 2001.
[129] M. A. Gibson and J. Bruck, "A probabilistic model of a prokaryotic gene and its regulation," in *Computational Modeling of Genetic and Biochemical Networks*, J. M. Bower and H. Bolouri, Eds., pp. 49–71, MIT Press, Cambridge, Mass, USA, 2001.
[130] M. Ginkel, A. Kremling, T. Nutsch, R. Rehner, and E. D. Gilles, "Modular modeling of cellular systems with ProMoT/Diva," *Bioinformatics*, vol. 19, no. 9, pp. 1169–1176, 2003.

[131] P. J. Goss and J. Peccoud, "Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 12, pp. 6750–6755, 1998.

[132] J. Hasty, F. Isaacs, M. Dolnik, D. McMillen, and J. J. Collins, "Designer gene networks: Towards fundamental cellular control," *Chaos*, vol. 11, no. 1, pp. 207–220, 2001.

[133] S. A. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, "Random Boolean network models and the yeast transcriptional network," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 25, pp. 14796–14799, 2003.

[134] L. Raeymaekers, "Dynamics of Boolean networks controlled by biologically meaningful functions," *J. Theor. Biol.*, vol. 218, no. 3, pp. 331–341, 2002.

[135] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, NY, USA, 1993.

[136] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, 1969.

[137] S. A. Kauffman, "Cellular homeostasis, epigenesis and replication in randomly aggregated macromolecular systems," *Journal of Cybernetics*, vol. 1, pp. 71–96, 1971.

[138] S. A. Kauffman, "Requirements for evolvability in complex systems: orderly dynamics and frozen components," *Phys. D*, vol. 42, pp. 135–152, 1990.

[139] R. Albert and H. G. Othmer, "The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*," *J. Theor. Biol.*, vol. 223, no. 1, pp. 1–18, 2003.

[140] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention," *J. Biol. Systems*, vol. 10, no. 4, pp. 431–445, 2002.

[141] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.

[142] J. Hasty, J. Pradines, M. Dolnik, and J. J. Collins, "Noise-based switches and amplifiers for gene expression," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 5, pp. 2075–2080, 2000.

[143] M. Tomita, K. Hashimoto, K. Takahashi, et al., "E-CELL: software environment for whole-cell simulation," *Bioinformatics*, vol. 15, no. 1, pp. 72–84, 1999.

[144] K. Takahashi, N. Ishikawa, Y. Sadamoto, et al., "E-Cell 2: multi-platform E-Cell simulation system," *Bioinformatics*, vol. 19, no. 13, pp. 1727–1729, 2003.

[145] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano, "Biopathways representation and simulation on hybrid functional Petri net," *In Silico Biol.*, vol. 3, no. 3, pp. 389–404, 2003.

[146] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano, "Hybrid Petri net representation of gene regulatory network," *Pac. Symp. Biocomput.*, vol. 5, pp. 341–352, 2000.

[147] M. Aldana and P. Cluzel, "A natural class of robust networks," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 15, pp. 8710–8714, 2003.

[148] S. A. Kauffman, "The large scale structure and dynamics of gene control circuits: an ensemble approach," *J. Theor. Biol.*, vol. 44, no. 1, pp. 167–190, 1974.

[149] I. Shmulevich, H. Lahdesmaki, E. R. Dougherty, J. Astola, and W. Zhang, "The role of certain Post classes in Boolean network models of genetic networks," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 19, pp. 10734–10739, 2003.

[150] S. Kim, H. Li, E. R. Dougherty, et al., "Can Markov chain mimic biological regulation?," *J. Biol. Systems*, vol. 10, no. 4, pp. 337–358, 2002.

[151] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Process.*, vol. 80, no. 10, pp. 2219–2235, 2000.

[152] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proc. IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[153] S. Kim, E. R. Dougherty, M. L. Bittner, et al., "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *J. Biomed. Opt.*, vol. 5, no. 4, pp. 411–424, 2000.

[154] S. Kim, E. R. Dougherty, Y. Chen, et al., "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.
[155] Y. Chen, E R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *J. Biomed. Opt.*, vol. 2, no. 4, pp. 364–374, 1997.
[156] M. L. Bittner, P. Meltzer, Y. Chen, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.
[157] A. T. Weeraratna, Y. Jiang, G. Hostetter, et al., "Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma," *Cancer Cell*, vol. 1, no. 3, pp. 279–288, 2002.
[158] R. F. Hashimoto, E. R. Dougherty, M. Brun, Z.-Z. Zhou, M. L. Bittner, and J. M. Trent, "Efficient selection of feature sets possessing high coefficients of determination based on incremental determinations," *Signal Process.*, vol. 83, no. 4, pp. 695–712, 2003.
[159] R. F. Hashimoto, S. Kim, I. Shmulevich, W. Zhang, M. L. Bittner, and E. R. Dougherty, "Growing genetic regulatory networks from seed genes," *Bioinformatics*, vol. 20, no. 8, pp. 1241–1247, 2004.
[160] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bittner, P. S. Meltzer, and J. M. Trent, "Ratio statistics of gene expression levels and applications to microarray data analysis," *Bioinformatics*, vol. 18, no. 9, pp. 1207–1215, 2002.
[161] S. B. Potterf, M. Furumura, K. J. Dunn, H. Arnheiter, and W. J. Pavan, "Transcription factor hierarchy in Waardenburg syndrome: regulation of MITF expression by SOX10 and PAX3," *Hum. Genet.*, vol. 107, no. 1, pp. 1–6, 2000.
[162] D. Lang and J. A. Epstein, "Sox10 and Pax3 physically interact to mediate activation of a conserved c-RET enhancer," *Hum. Mol. Genet.*, vol. 12, no. 8, pp. 937–945, 2003.
[163] N. Bondurand, V. Pingault, D. E. Goerich, et al., "Interaction among SOX10, PAX3 and MITF, three genes altered in Waardenburg syndrome," *Hum. Mol. Genet.*, vol. 9, no. 13, pp. 1907–1917, 2000.
[164] A. Ludwig, S. Rehberg, and M. Wegner, "Melanocyte-specific expression of dopachrome tautomerase is dependent on synergistic gene activation by the Sox10 and Mitf transcription factors," *FEBS Lett.*, vol. 556, no. 1-3, pp. 236–244, 2004.
[165] J. Du, A. J. Miller, H. R. Widlund, M. A. Horstmann, S. Ramaswamy, and D. E. Fisher, "MLANA/MART1 and SILV/PMEL17/GP100 are transcriptionally regulated by MITF in melanocytes and melanoma," *Am. J. Pathol.*, vol. 163, no. 1, pp. 333–343, 2003.

Seungchan Kim: Molecular Diagnostics and Target Validation Division, Translational Genomics Research Institute (TGen), Phoenix, AZ 85004, USA; Ira A. Fulton School of Engineering, Arizona State University, Tempe, AZ 85287, USA

Email: dolchan@tgen.org

Phillip Stafford: Computational Biology Division, Translational Genomics Research Institute (TGen), Phoenix, AZ 85004, USA; Arizona State University, Tempe, AZ 85287, USA

Email: pstafford@tgen.org

Michael L. Bittner: Molecular Diagnostics and Target Validation Division, Translational Genomics Research Institute (TGen), Phoenix, AZ 85004, USA

Email: mbittner@tgen.org

Edward B. Suh: High Performance Biocomputing and Bioinformatics Division, Translational Genomics Research Institute (TGen), Phoenix, AZ 85004, USA

Email: esuh@tgen.org