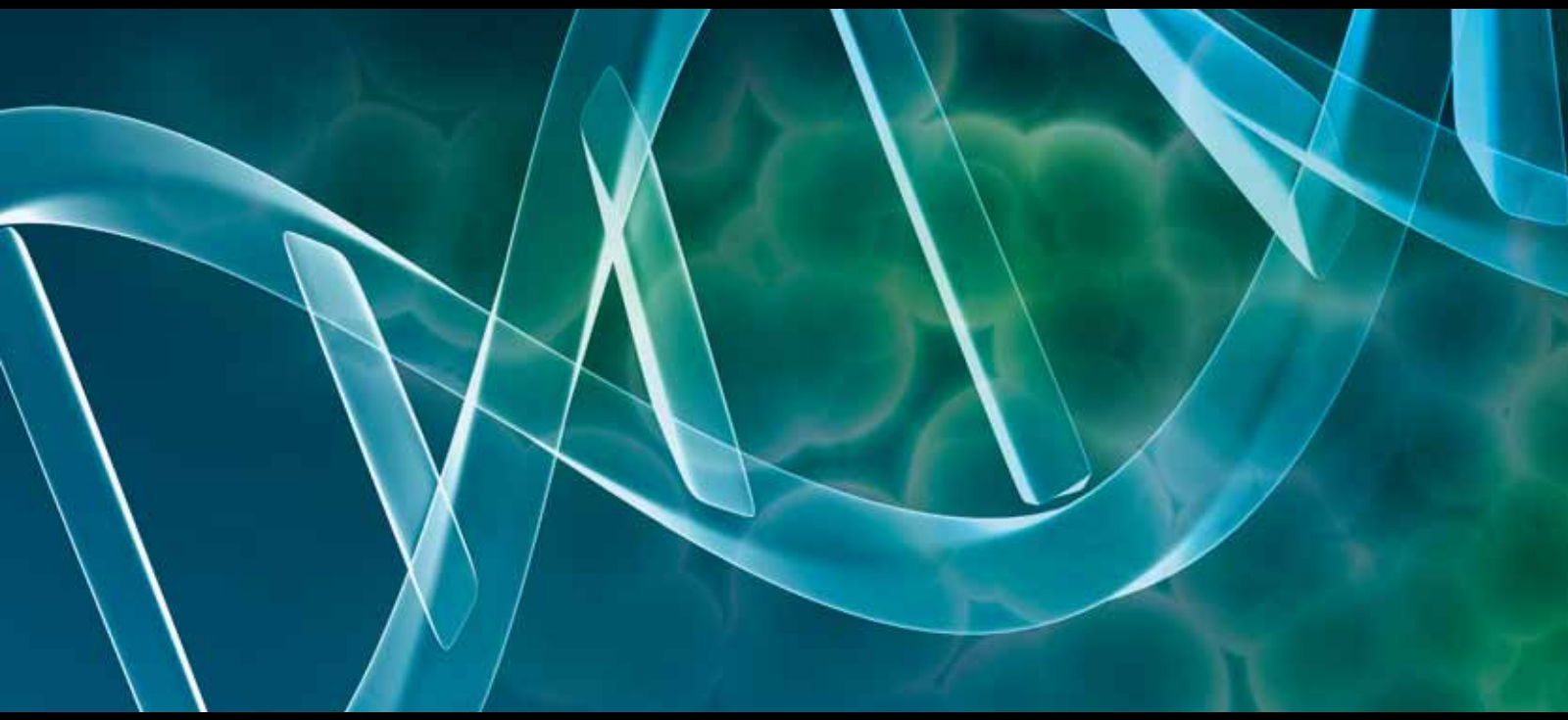
Advances in Bioinformatics

Computational and Statistical Approaches for Modeling of Proteomic and Genomic Networks

Guest Editors: Mohamed Nounou, Hazem Nounou, Erchin Serpedin, Aniruddha Datta, and Yufei Huang



Computational and Statistical Approaches for Modeling of Proteomic and Genomic Networks

Advances in Bioinformatics

Computational and Statistical Approaches for Modeling of Proteomic and Genomic Networks

Guest Editors: Mohamed Nounou, Hazem Nounou, Erchin Serpedin, Aniruddha Datta, and Yufei Huang

Editorial Board

Contents

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2013, Article ID 561968, 2 pages
http://dx.doi.org/10.1155/2013/561968

Editorial

Computational and Statistical Approaches for Modeling of Proteomic and Genomic Networks

Mohamed Nounou,[1] Hazem Nounou,[2] Erchin Serpedin,[3] Aniruddha Datta,[3] and Yufei Huang[4]

[1] *Chemical Engineering Program, Texas A&M University at Qatar, Doha, Qatar*
[2] *Electrical and Computer Engineering Program, Texas A&M University at Qatar, Doha, Qatar*
[3] *Electrical and Computer Engineering Department, Texas A&M University, College Station, TX 77843, USA*
[4] *Electrical and Computer Engineering Department, University of Texas at San Antonio, San Antonio, TX 78255, USA*

Correspondence should be addressed to Mohamed Nounou; mohamed.nounou@qatar.tamu.edu

Received 7 May 2013; Accepted 7 May 2013

Understanding and characterizing the behavior of biological systems triggered a huge interest among researchers to better understand how genes and proteins interact within cells by developing complex networks of structural, metabolic, and regulatory pathways. Advances in sensing technologies allowed collecting high throughput genomic and proteomic data that can be used in inferring the structure of proteomic and genomic networks. However, developing reliable algorithms for inferring genomic and proteomic networks and developing intervention techniques that can modify the behavior of biological systems are hindered by several factors. The most stringent limitations are the underdetermined nature of available data, which manifests in the large number of unknown variables and the small number of data samples, and the stochastic nature of the regulatory networks, which are often corrupted by noise and unknown latent variables during measurement. Therefore, developing computationally efficient data fusion, modeling, and intervention algorithms to overcome these limitations represents currently one of the most important research challenges in the field of computational biology. In this special issue, the authors have developed statistical and control techniques to model biological systems and to design intervention strategies that can help better understand these systems and can lead them to their more desirable states. This special issue consists of seven papers that address research topics along the lines mentioned previously. More details about the contributions of each paper in this special issue are presented next.

The paper "*Gene regulation, modulation, and their applications in gene expression data analysis*" by M. Flores et al. provided a unified mathematical description of the modulation of gene regulation, encompassing earlier mRNA expression-based methods and the more recent ceRNA method. The paper also presented applications to illustrate the construction of regulation networks, modulation effects, and the preliminary findings from these networks.

The paper "*Spectral analysis on time-course expression data: detecting periodic genes using a real-valued iterative adaptive approach*" by K. S. Agyepong et al. presented a novel scheme for detecting periodicities in time-course expression data using a real-valued iterative adaptive approach (RIAA), which is usually applied for periodogram estimation in signal processing. The spectrum obtained from spectral analysis is then analyzed using the Fisher's hypothesis test, and using a proper threshold, periodic genes can be detected. The detection scheme is illustrated through its application to simulated and real datasets.

The paper "*Identification of robust pathway markers for cancer through rank-based pathway activity inference*" by N. Khunlertgit and B.-J. Yoon presented an enhanced pathway activity inference method that uses gene ranking to predict the pathway activity in a probabilistic manner. This inference method is used to identify robust pathway markers that can ultimately lead to robust classifiers with reproducible performance across different genetic datasets. The advantages of the proposed method are illustrated through its application to breast cancer data.

The paper "*An overview of the statistical methods used for inferring gene regulatory networks and protein-protein interaction networks*" by A. Noor et al. provided a review of the most important statistical methods used for modeling gene regulatory networks (GRNs) and protein-protein interaction (PPI) networks. The paper focused on the recent advances in the statistical graphical modeling techniques, state-space representation models, and information theoretic methods that were proposed for inferring the topology of GRNs.

The paper "*MRMPath and MRMutation, facilitating discovery of mass transitions for proteotypic peptides in biological pathways using a bioinformatics approach*" by C. Crasto et al. described two software packages called MRMPath and MRMutation that are used to extract information from genomic data related to quantitative mass spectrometry analysis, such as the mass-to-charge ratio (m/z) values of proteotypic peptides and product ions. MRMPath utilizes publicly available information related to biological pathways from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database. MRMutation, on the other hand, catalogs and makes available, following processing, known (mutant) variants of proteins from the current UniProtKB database.

The paper "*Intervention in biological phenomena via feedback linearization*" by M. A. Fnaiech et al. presented an intervention technique to move biological systems from an undesirable state to a more desirable state. The authors considered biological phenomena represented by S-systems, and designed an intervention technique based on feedback linearization of the system model. The developed intervention technique is illustrated through its application to the glycolytic-glycogenolytic pathway model.

The paper "*Reverse engineering sparse gene regulatory networks using cubature Kalman filter and compressed sensing*" by A. Noor et al. presented a novel algorithm for inferring gene regulatory networks from time-series data. The algorithm makes use of the cubature Kalman filter (CKF) and the Kalman filter (KF) techniques in conjunction with compressed sensing methods to model the gene network as a state-space model. The developed algorithm is evaluated using simulated as well as real biological data sets, which include the DREAM4 *in silico* data sets and the *in vivo* data sets from the IRMA network.

Mohamed Nounou
Hazem Nounou
Erchin Serpedin
Aniruddha Datta
Yufei Huang

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2013, Article ID 205763, 11 pages
http://dx.doi.org/10.1155/2013/205763

Research Article

Reverse Engineering Sparse Gene Regulatory Networks Using Cubature Kalman Filter and Compressed Sensing

Amina Noor,[1] Erchin Serpedin,[1] Mohamed Nounou,[2] and Hazem Nounou[3]

[1] *Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA*
[2] *Chemical Engineering Department, Texas A&M University at Qatar, 253 Texas A&M Engineering Building, Education City, P.O. Box 23874, Doha, Qatar*
[3] *Electrical Engineering Department, Texas A&M University at Qatar, 253 Texas A&M Engineering Building, Education City, P.O. Box 23874, Doha, Qatar*

Correspondence should be addressed to Amina Noor; amina@neo.tamu.edu

Received 30 November 2012; Accepted 15 April 2013

Academic Editor: Yufei Huang

This paper proposes a novel algorithm for inferring gene regulatory networks which makes use of cubature Kalman filter (CKF) and Kalman filter (KF) techniques in conjunction with compressed sensing methods. The gene network is described using a state-space model. A nonlinear model for the evolution of gene expression is considered, while the gene expression data is assumed to follow a linear Gaussian model. The hidden states are estimated using CKF. The system parameters are modeled as a Gauss-Markov process and are estimated using compressed sensing-based KF. These parameters provide insight into the regulatory relations among the genes. The Cramér-Rao lower bound of the parameter estimates is calculated for the system model and used as a benchmark to assess the estimation accuracy. The proposed algorithm is evaluated rigorously using synthetic data in different scenarios which include different number of genes and varying number of sample points. In addition, the algorithm is tested on the DREAM4 *in silico* data sets as well as the *in vivo* data sets from IRMA network. The proposed algorithm shows superior performance in terms of accuracy, robustness, and scalability.

1. Introduction

Gene regulation is one of the most intriguing processes taking place in living cells. With hundreds of thousands of genes at their disposal, cells must decide which genes are to express at a particular time. As the cell development evolves, different needs and functions entail an efficient mechanism to turn the required genes on while leaving the others off. Cells can also activate new genes to respond effectively to environmental changes and perform specific roles. The knowledge of which gene triggers a particular genetic condition can help us ward off the potential harmful effects by switching that gene off. For instance, cancer can be controlled by deactivating the genes that cause it.

Gene expression is the process of generating functional gene products, for example, mRNA and protein. The level of gene functionality can be measured using microarrays or gene chips to produce the gene expression data [1]. More accurate estimation of gene expression is now possible using the RNA-Seq method. Intelligent use of such data can help improve our understanding of how the genes are interacting in a living organism [2–4]. Gene regulation is known to exhibit several modes; a couple of important ones include transcription regulation and posttranscription regulation [5]. While the theoretical applications of gene regulation are extremely promising, it requires a thorough understanding of this complex process. Different genes may cooperate to produce a particular reaction, while a gene may repress another gene as well. The potential benefits of gene regulation can only be reaped if a complete and accurate picture of genetic interactions is available. A network specifying different interconnections of genes can go a long way in understanding the gene regulation mechanism. The control and interaction of genes can be described through a *gene regulatory network*. Such a network depicts various interdependencies among genes where nodes of the network represent the genes,

and the edges between them correspond to an interaction among them. The strength of these interactions represents the extent to which a gene is affected by other genes in the network. A key ingredient of this approach is an accurate and representative modeling of gene networks. Precise modeling of a regulatory network coupled with efficient inference and intervention algorithms can help in devising personalized medicines and cures for genetic diseases [6].

Various methods for gene network modeling have been proposed recently in the literature with varying degrees of sophistication [7–10]. These techniques can be broadly classified as static and dynamic modeling schemes. Static modeling includes the use of correlation, statistical independence for clustering [11–13], and information theoretic criteria [14–16]. On the other hand, dynamic models provide an insight into the temporal evolution of gene expressions and hence yield a more quantitative prediction on gene network behavior [17–20]. In order to incorporate the stochasticity of gene expressions, statistical techniques have been applied [13]. A rich literature is also available on the Bayesian modeling of gene networks [21–26]. Promoted in part by the Bayesian methods, the state-space approach is a popular technique to model the gene networks [27–33], whereby the hidden states can be estimated using the Kalman filter. In the case of nonlinear functions, the extended Kalman filter (EKF) and particle filter represent feasible approaches [33, 34]. However, the EKF relies on the first-order linear approximations of nonlinearities, while the particle filter may be computationally too complex. A comprehensive review of these methods can be found in [35].

In this paper, the gene network is modeled using a state-space approach, and the cubature Kalman filter (CKF) is used to estimate the hidden states of the nonlinear model [36, 37]. The gene expressions are assumed to evolve following a sigmoid squash function, whereas a linear function is considered for the expression data. The noise is assumed to be Gaussian for both the state evolution and gene expression measurements. As the gene network is assumed sparse, any simple mean square error minimization technique will not suffice for the estimation of static parameters. Therefore, a compressed sensing-based Kalman filter (CSKF) [38] is used in conjunction with CKF for reliable estimation of parameters. In case of statistical inference, it is essential to obtain some guarantees on the performance of estimators. In this regard, the Cramér-Rao lower bound (CRB) of the parameter estimates is used as a benchmarking index to assess the mean square error (MSE) performance of the proposed estimator which is evaluated here for a parameter vector. The performance of the proposed algorithm is tested on synthetically generated random Boolean networks in various scenarios. The algorithm is also tested using DREAM4 data sets and IRMA networks [39, 40].

The main contributions of this paper can be summarized as follows.

(1) CKF is proposed for the estimation of states, and a compressed sensing-based Kalman filter is used for the estimation of system parameters. The genes are known to interact with few other genes only necessitating the use of sparsity constraint for more accurate estimation. The proposed algorithm carries out online estimation of parameters and is therefore computationally efficient and is particularly suitable for large gene networks.

(2) The Cramér-Rao lower bound is calculated for the estimation of unknown parameters of the system. The performance of the proposed algorithm is compared to CRB. This comparison is significant as it shows room for improvement in the estimation of parameters.

(3) The proposed algorithm is compared with the EKF algorithm. Using the false alarm errors, true connections, and Hamming distance as fidelity criteria, rigorous simulations are carried out to assess the performance of the algorithm with the increase in the number of samples. In addition, receiver operating characteristic (ROC) curves are plotted to evaluate the algorithms for different network sizes. It is observed that the proposed algorithm outperforms EKF in terms of accuracy and precision. The proposed algorithm is then applied to the DREAM4 10-gene and 100-gene data sets to assess the algorithm accuracy. The underlying gene network for the IRMA data sets is also inferred.

The rest of this paper is organized as follows. Section 2 describes the underlying system model for the gene expressions. The proposed CKF algorithm in combination with CSKF for gene network inference is formulated in Section 3. The derivation of CRB is shown in Section 4, and the simulation results and their interpretation are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. System Model

Gene regulatory networks can be modeled as static or dynamical systems. In this work, state-space modeling is considered which is an instance of a dynamic modeling approach and can effectively cope with time variations. The states represent gene expressions, and their evolution in time, in general, can be expressed as

$$\mathbf{x}_k = g\left(\mathbf{x}_{k-1}\right) + \mathbf{w}_k \quad k = 1, \ldots, K, \tag{1}$$

where K is the total number of data points available, $\mathbf{w}_k$ is assumed to be a zero-mean Gaussian random variable with covariance $\mathbf{Q}_k = \sigma_w^2\mathbf{I}$, and the function $g(\cdot)$ represents the regulatory relationship between the genes and is generally nonlinear. The microarray data is a set of noisy observations and is commonly expressed as a linear Gaussian model [41]

$$\mathbf{y}_k = h\left(\mathbf{x}_k\right) + \mathbf{v}_k, \tag{2}$$

where $\mathbf{v}_k$ is Gaussian-distributed random variable with zero mean and covariance $\mathbf{S}_k = \sigma_v^2\mathbf{I}$ and incorporates the uncertainty in the microarray experiments. In order to capture

the gene interactions effectively, the following nonlinear state evolution model is assumed [33, 34]:

$$x_{k,n} = \sum_{m=1}^{N} b_{nm} f\left(x_{k-1,m}\right) + w_{k,n}, \quad k = 1, \ldots, K, \; n = 1, \ldots, N, \tag{3}$$

where N is the total number of genes in the network and $f(\cdot)$ is the sigmoid squash function

$$f\left(x_{k-1,m}\right) = \frac{1}{1 + e^{-x_{k-1,m}}}. \tag{4}$$

This particular choice for the nonlinear function ensures that the conditional distribution of the states remains Gaussian [41]. The multiplicative constants b_{nm} quantify the positive or negative relations between various genes in the network. A positive value of b_{nm} implies that the mth gene is activating the nth gene, whereas a negative value implies repression [34, 42]. The absolute value of these parameters indicates the strength of interaction.

The model given in (3) and (4) in the absence of any constraints may be unidentifiable and may result into overfitted solutions [43]. Assumptions on network structures are, therefore, necessary to obtain a connectivity matrix that agrees with the biological knowledge. In a gene regulatory network (GRN), the genes are known to interact with few other genes only. To this end, the coefficients b_{nm}s are estimated using sparsity constraints, as explained in the next section.

A discrete linear Gaussian model for the microarray data is considered which can be expressed at the kth time instant as [41]

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{v}_k. \tag{5}$$

Stacking the unknown parameters together, the parameter vector to be estimated is

$$\mathbf{b} \triangleq \left[\phi_1, \phi_2, \ldots, \phi_N\right], \tag{6}$$

where $\phi_n = [b_{n1}, \ldots, b_{nN}]$. Plugging the values of states from (3) into (5), it follows that

$$\mathbf{y}_k = \mathbf{R}_k \mathbf{b} + \mathbf{e}_k, \tag{7}$$

where

$$\mathbf{R}_k \triangleq \begin{bmatrix} \widetilde{\mathbf{f}}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{f}}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{f}}_k \end{bmatrix}, \tag{8}$$

$$\widetilde{\mathbf{f}}_k \triangleq \left[f\left(x_{k-1,1}\right) \cdots f\left(x_{k-1,N}\right)\right]. \tag{9}$$

Thus, the gene network inference problem boils down to the estimation of system parameters $\mathbf{b}$ using the observations $\mathbf{y}_k$, where the effective noise $\mathbf{e}_k$ is the sum of system and observation noises. The next section describes the proposed inference algorithm for sparse networks.

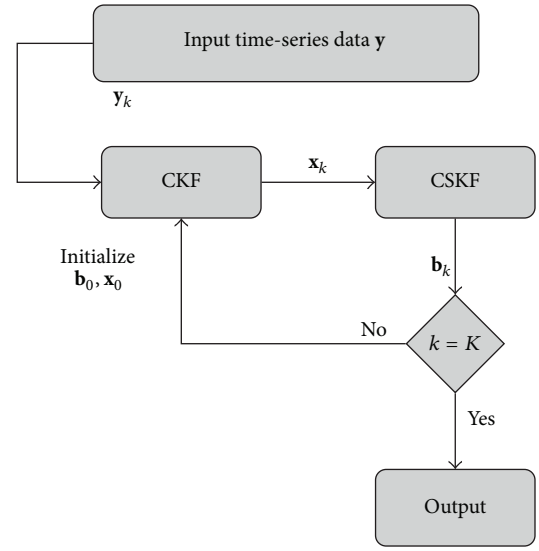


Figure 1: Block diagram of network inference methodology CKFS.

3. Method

In this section, the methodology proposed to infer the system parameters in (3) is described. The proposed cubature Kalman filter with sparsity constraints (CKFS) approach is succinctly illustrated in Figure 1. The specific details of this algorithm are as next presented.

3.1. Cubature Kalman Filter. Kalman filter is a Bayesian filter which provides the optimal solution to a general linear state space inference problem depicted by (1) and (2) and assumes a recursive *predictive-update* process. The underlying assumption of Gaussianity for the predictive and the likelihood densities simplifies the Kalman filter algorithm to a two-step process, consisting of prediction and update of the mean and covariance of the hidden states. However, the presence of nonlinear functions in the state and measurement equations requires calculation of multidimensional integrals of the form *nonlinear function* × *Gaussian density* [36], which in general is computationally prohibitive. Several solutions to this problem have been proposed including the EKF, which linearizes the nonlinear function by taking its first-order Taylor approximation, and the unscented Kalman filter (UKF), which approximates the probability density function (PDF) using a nonlinear transformation of the random variable. Recently, a new approach, CKF, has been proposed which evaluates the integrals numerically using spherical-radial cubature rules [36].

The next two subsections briefly explain the working of Bayesian filtering and the CKF solution for the nonlinear multidimensional integrals.

3.1.1. Time Update. Let the observations up to the time instant k be denoted by $\mathbf{d}_k$; that is, $\mathbf{d}_k \triangleq [\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T]^T$. In the prediction phase, also called the time update of the Bayesian filter,

the mean and covariance of the Gaussian posterior density are computed as follows:

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k-1} &= E\left[\mathbf{f}\left(\mathbf{x}_{k-1}\right) \mid \mathbf{d}_{k-1}\right],\\ \mathbf{P}_{xx,k|k-1} &= E\left[\mathbf{f}\left(\mathbf{x}_k\right)\mathbf{f}^T\left(x_k\right)\right] - \widehat{\mathbf{x}}_{k|k-1}\widehat{\mathbf{x}}_{k|k-1}^T + \mathbf{Q}_{k-1},\end{aligned} \tag{10}$$

where E denotes the expectation operator and $\mathbf{x}_{k-1}$ is normally distributed with parameters $(\widehat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{xx,k-1|k-1})$. The third equality is a consequence of the zero-mean nature of Gaussian noise $\mathbf{w}$ and its independence from $\mathbf{d}_k$. The estimates $\widehat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{xx,k-1|k-1}$ are assumed to be available from the previous iteration. Here, $\mathbf{P}_{xx,k|k-1}$ is an estimate of the error covariance matrix.

3.1.2. Measurement Update. Since the measurement noise is also Gaussian, the likelihood density is given by $\mathbf{y}_{k-1} \mid \mathbf{d}_{k-1} : \mathcal{N}(\mathbf{z}_{k-1}; \widehat{\mathbf{y}}_{k|k-1}, \mathbf{P}_{xx,k|k-1})$. As the measurements become available at the kth time instant, the mean and covariance of the likelihood density are calculated as follows:

$$\begin{aligned}\widehat{\mathbf{y}}_{k|k-1} &= E\left[\mathbf{y}_k \mid \mathbf{d}_{k-1}\right],\\ \mathbf{P}_{yy,k|k-1} &= E\left[\mathbf{x}_k\mathbf{x}_k^T\right] - \widehat{\mathbf{y}}_{k|k-1}\widehat{\mathbf{y}}_{k|k-1}^T + \mathbf{S}_{k-1}.\end{aligned} \tag{11}$$

The updated posterior density, obtained from the conditional joint density of states, and the measurements can be expressed as

$$\begin{aligned}&\left(\left[\mathbf{x}_k^T\mathbf{y}_k^T\right]^T\mathbf{d}_{\mathbf{k-1}}\right)\\ &\quad\sim \mathcal{N}\left(\begin{pmatrix}\widehat{\mathbf{x}}_{k|k-1}\\ \widehat{\mathbf{y}}_{k|k-1}\end{pmatrix}, \begin{pmatrix}\mathbf{P}_{xx,k|k-1} & \mathbf{P}_{xy,k|k-1}\\ \mathbf{P}_{xy,k|k-1}^T & \mathbf{P}_{yy,k|k-1}\end{pmatrix}\right),\end{aligned} \tag{12}$$

where

$$P_{xy,k|k-1} = E\left[\mathbf{x}_k\mathbf{x}_k^T\right] - \widehat{\mathbf{x}}_{k|k-1}\widehat{\mathbf{y}}_{k|k-1}^T \tag{13}$$

is the cross-covariance matrix between the states and the measurements. Hence, the states and the corresponding error covariance matrix are updated by calculating the innovation $\mathbf{z}_k - \widehat{\mathbf{z}}_{k|k-1}$ and the Kalman gain $\mathbf{K}_{G,i}$

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k} &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_{G,k}\left(\mathbf{y}_k - \widehat{\mathbf{y}}_{k|k-1}\right),\\ P_{xx,k|k} &= P_{xx,k|k-1} - \mathbf{K}_{G,k}P_{yy,k|k-1}\mathbf{K}_{G,k}^T,\\ \mathbf{K}_{G,k} &= P_{xy,k|k-1}P_{yy,k|k-1}^{-1}.\end{aligned} \tag{14}$$

The next subsection briefly describes the computation of high-dimensional integrals present in the equations above.

3.1.3. Computation of Integrals Using Spherical-Radial Cubature Points. In order to determine the expectations in (10), using a numerical integration method, a spherical-radial cubature rule is applied. This method calculates the cubature points $\mathbf{X}_{j,k-1|k-1}$ as follows [36]:

$$\mathbf{X}_{j,k-1|k-1} = \mathbf{U}_{k-1|k-1}\zeta_j + \widehat{\mathbf{x}}_{k-1|k-1}, \tag{15}$$

where $\zeta_j = \sqrt{\ell/2}[1]_j$, $j = 1, \ldots, \ell$, $\ell = 2N$ denotes the total number of cubature points and $\mathbf{U}_{k-1|k-1}$ stands for the square root of the error covariance matrix; that is,

$$\mathbf{P}_{xx,k-1|k-1} = \mathbf{U}_{k-1|k-1}\mathbf{U}_{k-1|k-1}^T. \tag{16}$$

The cubature points are updated via the state equation

$$\mathbf{X}_{j,k|k-1}^* = g\left(\mathbf{X}_{j,k-1|k-1}\right). \tag{17}$$

The propagated cubature points yield the state and error covariance estimates

$$\begin{aligned}\widehat{\mathbf{x}}_{k|k-1} &= \frac{1}{\ell}\sum_{j=1}^{\ell}\mathbf{X}_{j,k|k-1}^*,\\ \mathbf{P}_{xx,k|k-1} &= \frac{1}{\ell}\sum_{j=1}^{\ell}\mathbf{X}_{j,k|k-1}^*\mathbf{X}_{j,k|k-1}^{*T}\\ &\quad - \widehat{\mathbf{x}}_{k|k-1}\widehat{\mathbf{x}}_{k|k-1}^T + \mathbf{Q}_{k-1}.\end{aligned} \tag{18}$$

The integrals in (11) and (14) can be evaluated in a similar manner. The next subsection explains the estimation of parameters in the system.

3.2. Estimation of Sparse Parameters Using Kalman Filter. The state estimates are obtained using the CKF as described in the previous subsection. In order to estimate the unknown parameters in the system model, one of the most commonly used methods involves stacking the parameters with the states and estimating them together. The estimation process performed in this manner is called *joint estimation*. Another method for the estimation of parameters consists of a two-step recursive process which is termed *dual estimation*. This process estimates the states in the first step, and with the assumption that states are known, parameters are estimated in the second step. These steps are repeated until the algorithm converges to the true values or until the amount of available observations is exhausted. This paper makes use of the latter technique.

The vector $\mathbf{b}$ as defined in (6) is assumed to be evolving as a Gauss-Markov model. As discussed previously, the states are assumed to be known at this step. The system evolution equations can therefore be expressed as

$$\begin{aligned}\mathbf{b}_k &= \mathbf{b}_{k-1} + \boldsymbol{\eta}_{k-1},\\ \mathbf{y}_k &= \mathbf{R}_k\mathbf{b}_k + \mathbf{e}_k,\end{aligned} \tag{19}$$

where $\boldsymbol{\eta}_k$ stands for the i.i.d Gaussian noise and $\mathbf{R}_k$ is as defined in (8). It is observed that (19) is a system of linear equations with additive Gaussian noise, and therefore, the Kalman filter is the optimal choice for the estimation of

parameter vector. The standard *predict* and *update* steps involved in Kalman filter are summarized as follows:

$$\begin{aligned}
\widehat{\mathbf{b}}_{k|k-1} &= \widehat{\mathbf{b}}_{k-1|k-1} + \boldsymbol{\eta}_k,\\
\mathbf{P}_{bb,k|k-1} &= \mathbf{P}_{bb,k-1|k-1} + \boldsymbol{\Sigma}_{\eta_k},\\
\mathbf{u}_k &= \mathbf{y}_k - \mathbf{R}_{f_k}\widehat{\mathbf{b}}_k,\\
\mathbf{K}_G &= \mathbf{P}_{bb,k|k-1}\mathbf{R}_{f_k}^T\left(\mathbf{R}_{f_k}\mathbf{P}_{bb,k|k-1}\mathbf{R}_{f_k}^T + \sigma_e^2\mathbf{I}^{-1}\right),\\
\widehat{\mathbf{b}}_{k|k} &= \widehat{\mathbf{b}}_{k|k-1} + \mathbf{K}_G\mathbf{u}_k,\\
\mathbf{P}_{bb,k|k} &= \left(\mathbf{I} - \mathbf{K}_G\mathbf{R}_{f_k}\right)\mathbf{P}_{bb,k|k-1},
\end{aligned} \tag{20}$$

where $\mathbf{K}_G$ denotes the Kalman gain and $\mathbf{P}$ represents the error covariance matrix.

The Kalman filter algorithm is based on an l_2-norm minimization criterion. As the gene networks are known to be highly sparse, the parameter vector is expected to have only a few nonzero values. A more accurate approach for estimating such a vector would be to introduce an additional constraint on its l_1-norm which is the core idea in compressed sensing [38, 44]. Such an l_1-norm constraint provides a unique solution to the underdetermined set of equations [45]. Therefore, instead of a simple l_2 norm minimization, the following constrained optimization problem is considered:

$$\min_{\widehat{\mathbf{b}}_k}\left\|\widehat{\mathbf{b}}_k - \mathbf{b}_k\right\|_2^2 \quad \text{s.t. } \left\|\widehat{\mathbf{b}}_k\right\| \le \epsilon. \tag{21}$$

The importance of this constraint can be judged by the fact that without it, the system would be rendered unidentifiable [43].

The problem (21) can be solved using a pseudomeasurement (PM) method which incorporates the inequality constraint (21) in the filtering process by assuming an artificial measurement $\|\mathbf{b}_k\|_1 - \epsilon = 0$. This is concisely expressed as

$$0 = \overline{\mathbf{R}}\widehat{\mathbf{b}}_k - \epsilon, \quad \overline{\mathbf{R}}_\tau = \left[\operatorname{sign}\left(\widehat{\mathbf{b}}_\tau(1)\right), \ldots, \operatorname{sign}\left(\widehat{\mathbf{b}}_\tau(N)\right)\right]. \tag{22}$$

The value of the covariance matrix $\boldsymbol{\Sigma}_\epsilon = \sigma_\epsilon^2\mathbf{I}$ of the pseudonoise ϵ is selected in a similar manner as the process noise covariance in the EKF algorithm. However, it is found that large values of variances, that is, $\sigma_\epsilon^2 \ge 100$, prove sufficient in most cases [38]. Further details on selecting these parameters can be found in [38, 46]. The PM method solves (21) in a recursive manner for K_τ iterations using the following steps:

$$\begin{aligned}
\mathbf{K}_G^\tau &= \mathbf{P}_\tau\overline{\mathbf{R}}_\tau^T\left(\overline{\mathbf{R}}_\tau\mathbf{P}_\tau\overline{\mathbf{R}}_\tau^T + \boldsymbol{\Sigma}_\epsilon\right)^{-1},\\
\widehat{\mathbf{b}}_{\tau+1} &= \left(\mathbf{I} - \mathbf{K}_G^\tau\overline{\mathbf{R}}_\tau\right)\widehat{\mathbf{b}}_\tau,\\
\mathbf{P}_{\tau+1} &= \left(\mathbf{I} - \mathbf{K}_G^\tau\overline{\mathbf{R}}_\tau\right)\mathbf{P}_\tau.
\end{aligned} \tag{23}$$

At each kth time instant, $\mathbf{P}_{bb,k|k}$ and $\widehat{\mathbf{b}}_{k|k}$ obtained from (20) are considered as initial values; that is, $\widehat{\mathbf{b}}^1 = \widehat{\mathbf{b}}_{k|k}$ and $\mathbf{P}_1 = \mathbf{P}_{bb,k|k}$ which is the error covariance matrix. The value of K_τ is equal to the number of constraints, that is, the expected number of nonzero $\mathbf{b}_{mn}$s in the system. Possible ways for calculating K_τ include minimum description length (MDL) principle and Bayesian information criterion (BIC).

(1) Input time series data set $\mathbf{y}$.
(2) Initialize $I, K, \phi_0, \mathbf{x}_0$.
(3) **for** $k = 1, \ldots, K$ **do**
(4) Find the state estimates using CKF following the time and measurement update steps in Section 3.
(5) Estimate parameters $\widehat{\mathbf{b}}_k$ from $\mathbf{x}_k$ and $\mathbf{y}_k$ using (20).
(6) **for** $\tau = 1, \ldots, K_\tau$ **do**
(7) Update the parameters $\widehat{\mathbf{b}}_k$ using (23).
(8) **end for**
(9) **end for**
(10) **return**

Algorithm 1: Network inference: CKFS.

3.3. Inference Algorithm. The network inference algorithm is summarized in Algorithm 1. The algorithm consists of a recursive process which repeats itself for the number of observations present in the time-series data. For each time sample, the state estimate is obtained using the CKF, and the parameter estimate is obtained using the KF. Since the parameters are expected to be sparse, the estimates are then refined further using the CSKF algorithm. This iterative process results in a simple and accurate algorithm for gene network inference while considering a complex nonlinear model.

4. Cramér-Rao Bound

The performance of an estimator can be judged by comparing it with theoretical lower bounds proposed in parameter estimation theory. The CRB establishes a lower bound on the MSE of an unbiased estimator [47]. In particular, the CRB states that the covariance matrix of the estimator $\widehat{\mathbf{b}}$ is lower bounded by

$$\mathbb{E}\left[\left(\widehat{\mathbf{b}} - \mathbf{b}\right)\left(\widehat{\mathbf{b}} - \mathbf{b}\right)^T\right] \succeq \left[\mathbf{I}(\mathbf{b})\right]^{-1}, \tag{24}$$

where the matrix inequality $\succeq$ is to be interpreted in the semidefinite sense and $\mathbf{I}(\mathbf{b})$ is the Fisher information matrix (FIM)

$$\mathbf{I}(\mathbf{b}) = \mathbb{E}\left[\left(\frac{\partial \ln f(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}}\right)\left(\frac{\partial \ln f(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}}\right)^T\right]. \tag{25}$$

The CRB for gene network inference can be calculated as follows. By stacking all the observations for $k = 1, \ldots, K$, (7) can be written compactly in the matrix form

$$\mathbf{y} = \mathbf{R}\mathbf{b} + \mathbf{e}, \tag{26}$$

where $\mathbf{y} = [\mathbf{y}_1^T, \ldots, \mathbf{y}_K^T]^T$, $\mathbf{R} = [\mathbf{R}_1^T, \ldots, \mathbf{R}_K^T]^T$, and $\mathbf{e} = [\mathbf{e}_1^T, \ldots, \mathbf{e}_K^T]^T$. The PDF $p(\mathbf{y} \mid \mathbf{b})$ is expressed as

$$p(\mathbf{y} \mid \mathbf{b}) = C \exp\left(-\frac{(\mathbf{y} - \mathbf{Rb})^T (\mathbf{y} - \mathbf{Rb})}{2\sigma_e^2}\right), \quad (27)$$

where C is a constant. The derivative of $\ln p(\mathbf{y} \mid \mathbf{b})$ can be expressed as

$$\frac{\partial \ln p(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}} = -\frac{\partial}{\partial \mathbf{b}}\left[\frac{(\mathbf{y} - \mathbf{Rb})^T (\mathbf{y} - \mathbf{Rb})}{\sigma_e^2}\right] = \frac{\mathbf{R}^T\mathbf{y} - \mathbf{R}^T\mathbf{Rb}}{\sigma_e^2}. \quad (28)$$

It now follows that

$$\left(\frac{\partial \ln p(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}}\right)\left(\frac{\partial \ln p(\mathbf{y} \mid \mathbf{b})}{\partial \mathbf{b}}\right)^T = \frac{\mathbf{R}^T (\mathbf{y} - \mathbf{Rb})(\mathbf{y} - \mathbf{Rb})^T \mathbf{R}}{\sigma_e^4}. \quad (29)$$

By taking the expectation of (29), the FIM in (25) is given by

$$\mathbf{I}(\mathbf{b}) = \frac{\mathbf{R}^T\mathbf{R}}{\sigma_e^2}. \quad (30)$$

The inverse of the FIM in (30) can be used to place a lower bound on the estimation error of the parameter vector $\mathbf{b}$. Figure 2 shows the comparison of MSE of CKFS algorithm with CRB as a function of number of samples K for one representative gene from the eight-gene network considered in Section 5.1. It is observed that the MSE of the estimated parameters decreases with increasing number of samples.

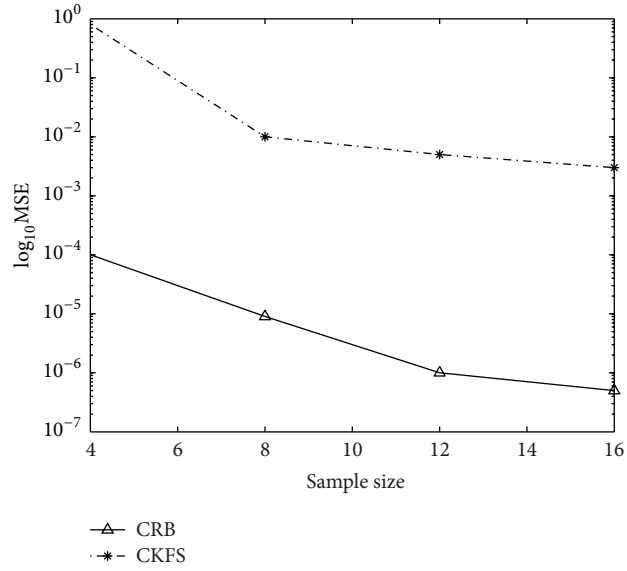


Figure 2: Cramér-Rao bound on the estimation of parameters. The MSE for one of the representative θ is shown here for a network consisting of 8 vertices.

5. Results and Discussion

The simulation results of the CKFS algorithm are discussed in this section. The performance is first tested on synthetic data obtained from randomly generated Boolean networks under various scenarios and performance metrics. The algorithm is then assessed on the DREAM4 networks and the IRMA network.

5.1. Synthetic Data. Time-series data is produced from randomly generated Boolean networks using the system model (3) and (5). Two scenarios are considered for this purpose.

First, the comparison is performed by varying the number of sample size while keeping the network size fixed. The gene network consists of 8 genes and 20 vertices. In terms of network estimation, if the algorithm predicts an edge between two nodes which may not be present in reality, an error, referred to as *false alarm error* (F), is said to have occurred. Another situation is the indication of the absence of a vertex in the graph which in fact is present in the real network. This kind of error is termed *missed detection* (M). The summation of these two errors normalized over the total number of vertices in the network yields the *Hamming distance*. It is also important to consider the probability of predicting the true connections correctly which will be assessed by the *true connections* (T) metric. An algorithm with low Hamming distance and small false alarm error is particularly desirable as predicting an edge erroneously can be troublesome for biologists. True connections indicate the reliability of the predictions. Figure 3 illustrates the performance of the CKFS algorithm and that of the EKF algorithm proposed in [34] in terms of the metrics described above. It is important to mention here that the same system model is assumed by both CKFS and EKF algorithms for the purpose of this simulation. These metrics are the same as those used in [15]. The variances of both the system and measurement noises, σ_w^2 and σ_v^2, respectively, are taken to be 10^{-5} in all the simulations and are assumed to be known. It is noticed that EKF has a slightly lower false alarm rate when the number of samples is small; however, as the number of samples increases, CKFS yields a lower false alarm error. The Hamming distance for CKFS is also smaller than EKF indicating lesser cumulative error. True connections show a consistent behavior for the two algorithms when the number of samples is increased where CKFS is able to predict connections more accurately. These experiments show the superiority of CKFS in terms of lower error rate.

To obtain a more rigorous evaluation, the performance of algorithms is then compared in a scenario which considers the sample size to be fixed and assumes networks of different sizes. The receiver operating characteristic (ROC) curves are plotted as performance measures. A higher area under the ROC curve (AUROC) shows more true positives for a given false positive, and therefore, indicates better classification. The performance of CKFS(N, E, K) and EKF(N, E, K) is shown in Figure 4, where N stands for the number of nodes,

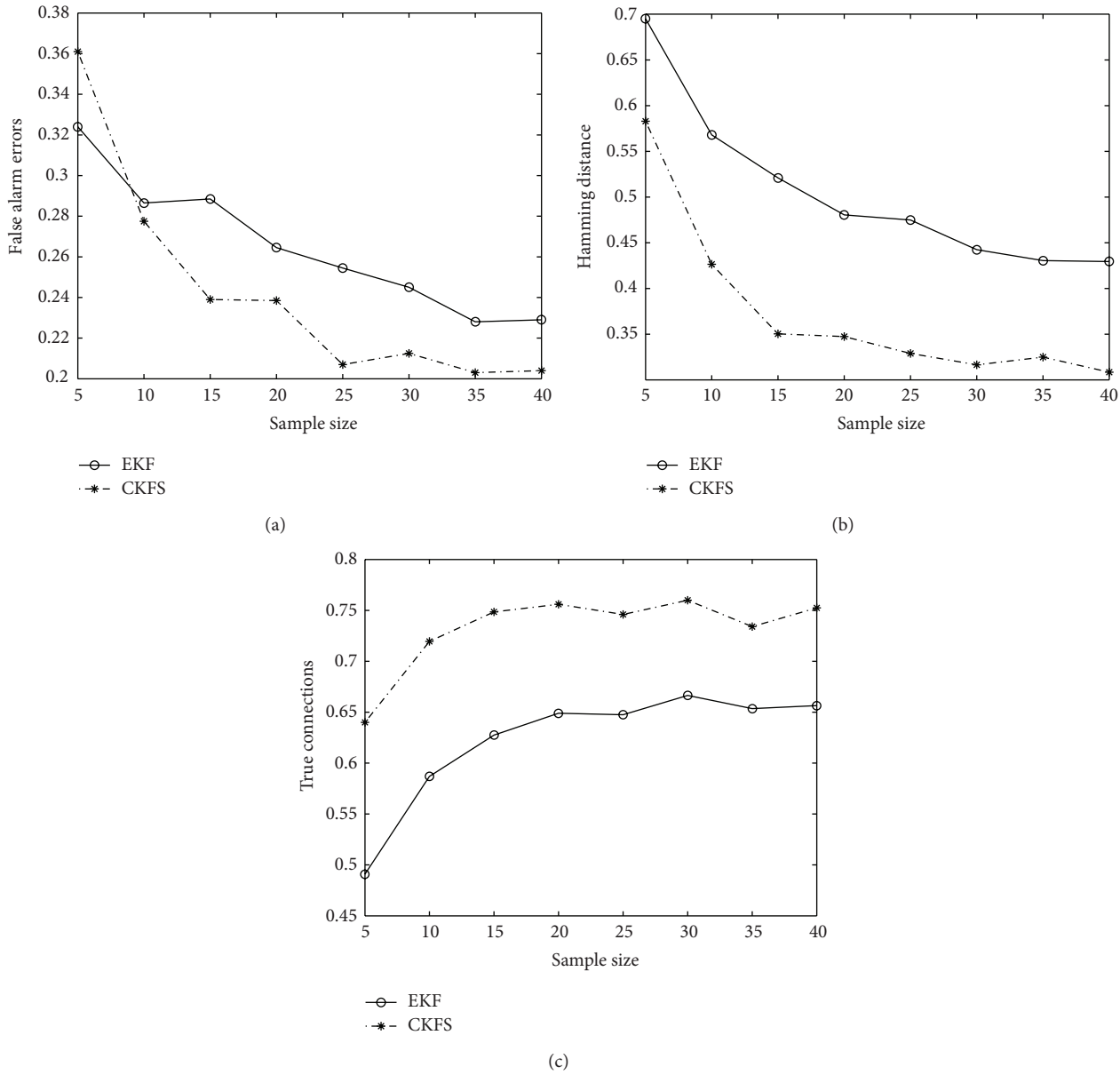


Figure 3: (a), (b), and (c) False alarm errors, Hamming distance, and true connections. The synthetic networks consist of 8 vertices and 20 edges. The metric is normalized over the number of edges. CKFS gives lower error and predicts more true connections with the increase in the sample size of data.

E represents the number of edges, and K denotes the time points. It is observed that the CKFS exhibits superior performance than the EKF for networks of different sizes.

The complexity of the two algorithms is compared for synthetically generated networks with number of genes equal to 10, 20, 30, and 40. The sample size is kept to 50 time points for each of these networks, and the run time for EKF and CKFS algorithms is calculated as shown in Table 1. It is noted that EKF is faster for smaller network sizes, but as the network size increases, the run time gets much larger than that for CKFS. The main reason for this is that EKF [34] estimates the states and parameters by stacking them together which requires large-sized matrix multiplications at each iteration. The benefit associated with performing dual estimation, as in CKFS, is that the parameters are estimated separately from the states. Since the system is linear and one-to-one for parameters, inversion of much smaller matrices can be performed reducing the computational complexity of CKFS algorithm. CKFS is therefore particularly attractive for large-sized networks.

5.2. DREAM4 Gene Networks. Several *in silico* networks have been produced in order to benchmark the performance of gene network inference algorithms. dialogue on reverse engineering assessment and methods (DREAM) *in silico* networks serve as one of the popular methods used for this

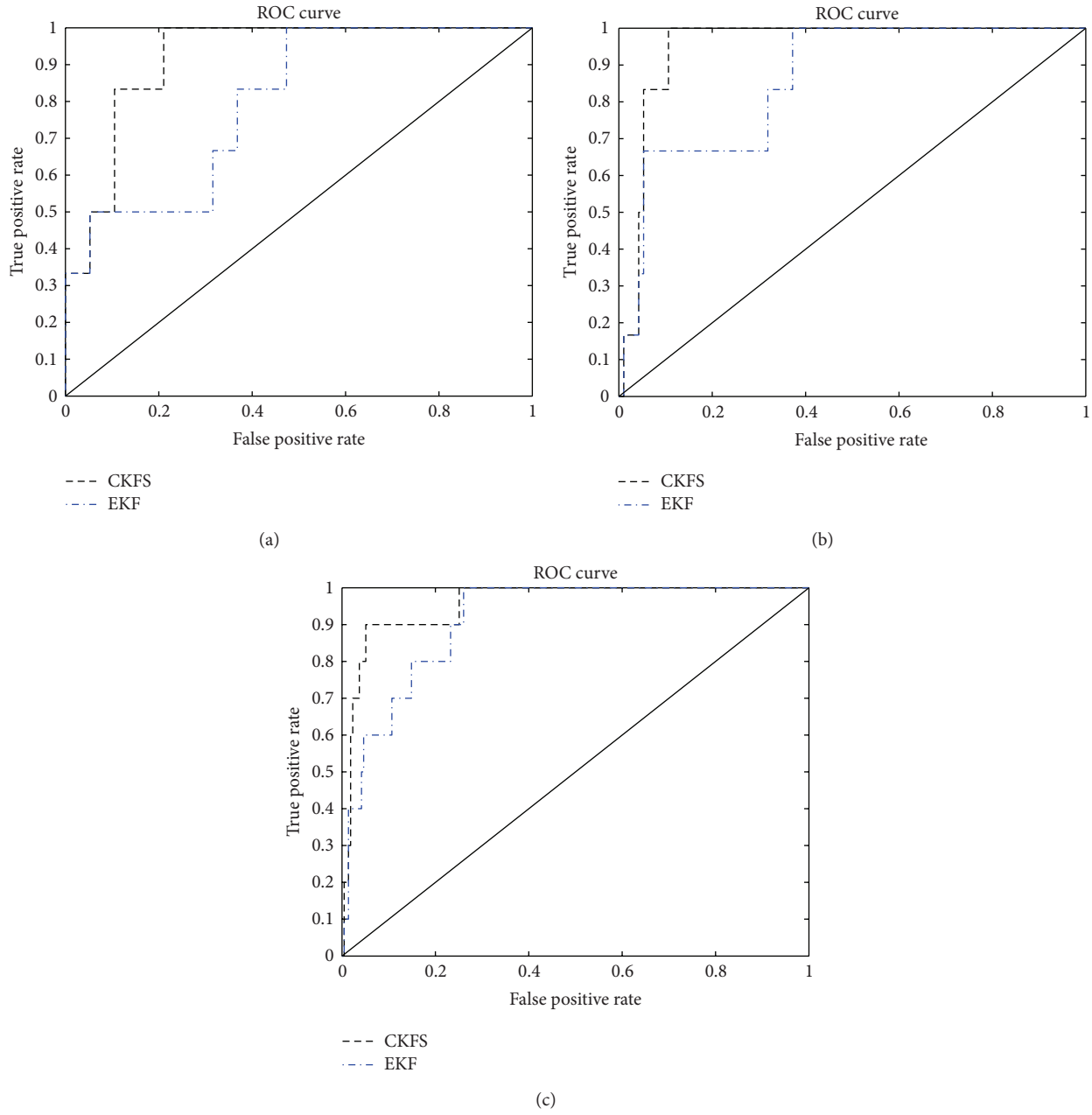


Figure 4: ROC curves for the performance of CKFS and EKF using synthetic data. (N, E, K) (a), (b), and (c) $(5, 10, 20)$, $(10, 12, 20)$, and $(15, 19, 20)$. The area under the ROC curve for CKFS is more than that for EKF for various sized networks.

Table 1: Run time in seconds for EKF and CKFS algorithms for varying network sizes for synthetically generated data. The number of sample points is fixed to 50.

Number of genes	10	20	30	40
EKF	0.16	1.9	16.5	84
CKFS	1.2	4.3	11.5	24.1

purpose [39, 48]. In this section, the performance of the CKFS algorithm is evaluated using the 10-gene and 100-gene networks released online by the DREAM4 challenge. Five networks are produced using the known GRNs of *Escherichia coli* and *Saccharomyces cerevisiae*. The data sets for each of 10-gene network consists of 21 data points for five different perturbations. The inference is performed by using all the perturbations. The 100-gene network consists of data sets for ten perturbations. AUROC and area under the precision-recall curve (AUPR) are calculated for the five networks of both the data sets and shown in Tables 2 and 3, respectively. The quantities, *precision* and *recall*, are defined as $P = T/(T + F)$ and $R = T/(T + M)$, respectively. For comparison purposes, the values of the two quantities for time-series network identification (TSNI) algorithm that

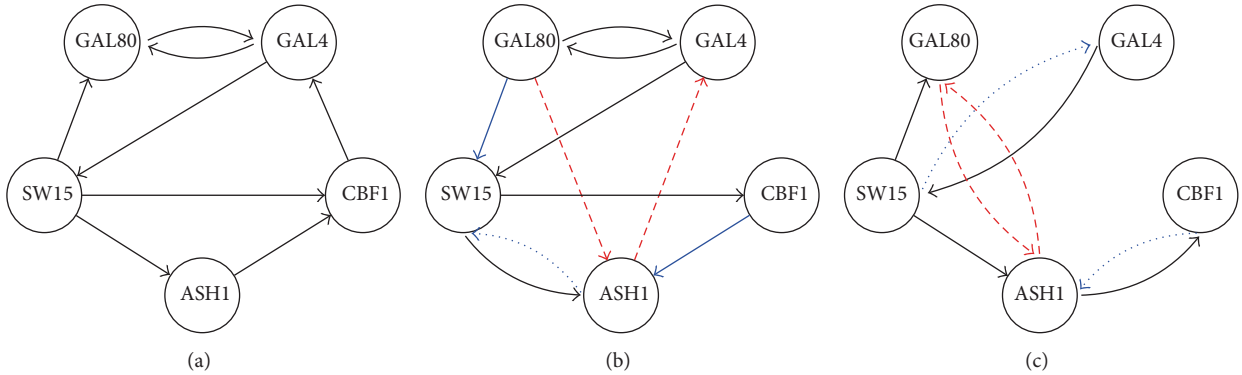


Figure 5: The inferred IRMA networks. (a), (b), and (c) Gold standard, inferred network using CKFS, and inferred network using ODE [39, 40]. Black arrows indicate true connections, blue arrows indicate the edges that are correct, but their directions are reversed, and red arrows indicate false positives.

Table 2: Area under the ROC curve (AUROC) and area under the PR curve (AUPR) for DREAM4 10-gene networks for the five different networks.

Algorithm	Network 1	Network 2	Network 3	Network 4	Network 5
ODE [39]	0.62 (0.27)	0.63 (0.32)	0.58 (0.21)	0.63 (0.23)	0.68 (0.25)
CKFS	0.63 (0.40)	0.67 (0.50)	0.72 (0.50)	0.75 (0.49)	0.81 (0.42)
Random [39]	0.55 (0.18)	0.55 (0.19)	0.55 (0.17)	0.57 (0.17)	0.56 (0.16)

Table 3: Area under the ROC curve (AUROC) and area under the PR curve (AUPR) for DREAM4 100-gene networks for the five different networks.

Algorithm	Network 1	Network 2	Network 3	Network 4	Network 5
ODE [39]	0.55 (0.02)	0.55 (0.03)	0.60 (0.03)	0.54 (0.02)	0.59 (0.03)
CKFS	0.67 (0.13)	0.57 (0.08)	0.60 (0.10)	0.62 (0.10)	0.60 (0.07)
Random [39]	0.50 (0.002)	0.50 (0.002)	0.50 (0.002)	0.50 (0.002)	0.50 (0.002)

exploits ordinary differential equations are also given [39]. The CKFS algorithm is found to perform significantly better than the TSNI algorithm.

5.3. IRMA Gene Network. In addition to synthetic data, it is imperative to test the algorithms using real biological data. In this subsection, the performance of the CKFS algorithm is assessed using the *in vivo* reverse-engineering and modeling assessment (IRMA) network [40]. This network consists of five genes. Galactose activates the gene expression in the network, whereas glucose deactivates it. The cells are grown in the presence of galactose and then switched to glucose to obtain the switch-off data which represents the expressive samples at 21 time points. The switch-on data consists of 16 sample points and is obtained by growing the cells in a glucose medium and then changing to galactose. The system and measurement noise variances for the CKFS are assumed to be identical as in the previous simulations. Figure 5 shows the inferred network, the gold standard, and the network inferred using TSNI. It is observed that the CKFS algorithm succeeds to predict most of the interactions while giving lower false positives.

6. Conclusions

This paper presents a novel algorithm for inferring gene regulatory networks from time-series data. Gene regulation is assumed to follow a nonlinear state evolution model. The parameters of the system, which indicate the inhibitory or excitatory relationships between the genes, are estimated using compressed sensing-based Kalman filtering. The sparsity constraint on the parameters is crucial because the genes are known to interact with few other genes only. The use of CKF and the dual estimation of states and parameters renders the algorithm computationally efficient. The performance of CKFS is evaluated for synthetic data for different network sizes as well as varying sample points. ROC curves, Hamming distance, and true positives are used for comparing the accuracy of inferred network with EKF. It is observed that CKFS outperforms the EKF algorithm. In addition, CKFS

gives advantages over EKF in terms of smaller run time for large networks. The Cramér-Rao lower bound is also determined for the parameters of the model and compared with the MSE performance of the proposed algorithm. Assessment using DREAM4 10-gene and 100-gene networks and IRMA network data corroborates the superior performance of CKFS. Future research directions include incorporating the estimation of model order in the network inference algorithm.

Acknowledgments

This work was supported by US National Science Foundation (NSF) Grant 0915444 and QNRF-NPRP Grant 09-874-3-235. The material in this paper was presented in part at the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS), San Antonio, TX, USA, December 2011.

References

[1] X. Zhou, X. Wang, and E. R. Dougherty, *Genomic Networks: Statistical Inference from Microarray Data*, John Wiley & Sons, New York, NY, USA, 2006.

[2] H. Kitano, "Computational systems biology," *Nature*, vol. 420, pp. 206–210, 2002.

[3] X. Zhou and S. T. C. Wong, *Computational Systems Bioinformatics*, World Scientific, River Edge, NJ, USA, 2008.

[4] X. Cai and X. Wang, "Stochastic modeling and simulation of gene networks," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 27–36, 2007.

[5] D. Yue, J. Meng, M. Lu, C. L. P. Chen, M. Guo, and Y. Huang, "Understanding micro-RNA regulation: a computational perspective," *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 77–88, 2012.

[6] R. Pal, S. Bhattacharya, and M. U. Caglar, "Robust approaches for genetic regulatory network modeling and intervention: a review of recent advances," *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 66–76, 2012.

[7] H. Hache, H. Lehrach, and R. Herwig, "Reverse engineering of gene regulatory networks: a comparative study," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2009, Article ID 617281, 2009.

[8] T. Schlitt and A. Brazma, "Current approaches to gene regulatory network modelling," *BMC Bioinformatics*, vol. 8, no. 6, p. 9, 2007.

[9] H. D. Jong, "Modeling and simulation of genetic regulatoy systems: a literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, 2002.

[10] I. Nachman, A. Regev, and N. Friedman, "Inferring quantitative models of regulatory networks from expression data," *Bioinformatics*, vol. 20, no. 1, pp. i248–i256, 2004.

[11] C. D. Giurcaneanu, I. Tabus, and J. Astola, "Clustering time series gene expression data based on sum-of-exponentials fitting," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 8, Article ID 358568, pp. 1159–1173, 2005.

[12] C. D. Giurcaneanu, I. Tabus, J. Astola, J. Ollila, and M. Vihinen, "Fast iterative gene clustering based on information theoretic criteria for selecting the cluster structure," *Journal of Computational Biology*, vol. 11, no. 4, pp. 660–682, 2004.

[13] X. Cai and G. B. Giannakis, "Identifying differentially expressed genes in microarray experiments with model-based variance estimation," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2418–2426, 2006.

[14] X. Zhou, X. Wang, and E. R. Dougherty, "Gene clustering based on cluster-wide mutual information," *Journal of Computational Biology*, vol. 11, no. 1, pp. 151–165, 2004.

[15] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring connectivity of genetic regulatory networks using informationtheoretic criteria," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 2, pp. 262–274, 2008.

[16] J. Dougherty, I. Tabus, and J. Astola, "Inference of gene regulatory networks based on a universal minimum description length," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2008, Article ID 482090, 2008.

[17] L. Qian, H. Wang, and E. R. Dougherty, "Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3327–3339, 2008.

[18] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring gene regulatory networks from time series data using the minimum description length principle," *Bioinformatics*, vol. 22, no. 17, pp. 2129–2135, 2006.

[19] 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.

[20] J. Meng, M. Lu, Y. Chen, S.-J. Gao, and Y. Huang, "Robust inference of the context specific structure and temporal dynamics of gene regulatory network," *BMC Genomics*, vol. 11, no. 3, p. S11, 2010.

[21] Y. Zhang, Z. Deng, H. Jiang, and P. Jia, "Inferring gene regulatory networks from multiple data sources via a dynamic Bayesian network with structural em.," in *DILS*, S. C. Boulakia and V. Tannen, Eds., vol. 4544 of *Lecture Notes in Computer Science*, pp. 204–214, Springer, New York, NY, USA, 2007.

[22] K. Murphy and S. Mian, Modeling gene expression data using dynamic Bayesian networks, University of California, Berkeley, Calif, USA, 2001.

[23] H. Liu, D. Yue, L. Zhang, Y. Chen, S. J. Gao, and Y. Huang, "A Bayesian approach for identifying miRNA targets by combining sequence prediction and gene expression profiling," *BMC Genomics*, vol. 11, no. 3, p. S12, 2010.

[24] Y. Huang, J. Wang, J. Zhang, M. Sanchez, and Y. Wang, "Bayesian inference of genetic regulatory networks from time series microarray data using dynamic Bayesian networks," *Journal of Multimedia*, vol. 2, no. 3, pp. 46–56, 2007.

[25] 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, no. 2, pp. ii138–ii148, 2003.

[26] C. Rangel, D. L. Wild, F. Falciani, Z. Ghahramani, and A. Gaiba, "A. modelling biological responses using gene expression profiling and linear dynamical systems," *Bioinformatics*, pp. 349–356, 2005.

[27] M. Quach, N. Brunel, and F. d'Alch Buc, "Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference," *Bioinformatics*, vol. 23, no. 23, pp. 3209–3216, 2007.

[28] F.-X. Wu, W.-J. Zhang, and A. J. Kusalik, "Modeling gene expression from microarray expression data with state-space

equations," in *Pacific Symposium on Biocomputing*, R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung, and T. E. Klein, Eds., pp. 581–592, World Scientific, River Edge, NJ, USA, 2004.

[29] R. Yamaguchi, S. Yoshida, S. Imoto, T. Higuchi, and S. Miyano, "Finding module-based gene networks with state-space modelsmining high-dimensional and short time-course gene expression data," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 37–46, 2007.

[30] O. Hirose, R. Yoshida, S. Imoto et al., "Statistical inference of transcriptional module-based gene networks from time course gene expression profiles by using state space models," *Bioinformatics*, vol. 24, no. 7, pp. 932–942, 2008.

[31] J. Angus, M. Beal, J. Li, C. Rangel, and D. Wild, "Inferring transcriptional networks using prior biological knowledge and constrained state-space models," in *Learning and Inference in Computational Systems Biology*, N. Lawrence, M. Girolami, M. Rattray, and G. Sanguinetti, Eds., pp. 117–152, MIT Press, Cambridge, UK, 2010.

[32] C. Rangel, J. Angus, Z. Ghahramani et al., "Modeling T-cell activation using gene expression profiling and state-space models," *Bioinformatics*, vol. 20, no. 9, pp. 1361–1372, 2004.

[33] A. Noor, E. Serpedin, M. N. Nounou, and H. N. Nounou, "Inferring gene regulatory networks via nonlinear state-space models and exploiting sparsity," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1203–1211, 2012.

[34] Z. Wang, X. Liu, Y. Liu, J. Liang, and V. Vinciotti, "An extended kalman filtering approach to modeling nonlinear dynamic gene regulatory networks via short gene expression time series," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 3, pp. 410–419, 2009.

[35] A. Noor, E. Serpedin, M. Nounou, H. Nounou, N. Mohamed, and L. Chouchane, "An overview of the statistical methods used for inferring gene regulatory networks and proteinprotein interaction networks," *Advances in Bioinformatics*, vol. 2013, Article ID 953814, 12 pages, 2013.

[36] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[37] A. Noor, E. Serpedin, M. N. Nounou, and H. N. Nounou, "A cubature Kalman filter approach for inferring gene regulatory networks using time series data," in *Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS '11)*, pp. 25–28, 2011.

[38] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using kalman filtering with embedded rseudo-measurement norms and quasi-norms," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2405–2409, 2010.

[39] C. A. Penfold and D. L. Wild, "How to infer gene networks from expression profiles, revisited," *Interface Focus*, pp. 857–870, 2011.

[40] I. Cantone, L. Marucci, F. Iorio et al., "A yeast synthetic network for *in vivo* assessment of reverse-engineering and modeling approaches," *Cell*, vol. 137, no. 1, pp. 172–181, 2009.

[41] Y. Huang, I. M. Tienda-Luna, and Y. Wang, "Reverse engineering gene regulatory networks: a survey of statistical models," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 76–97, 2009.

[42] Z. Wang, F. Yang, D. W. C. Ho, S. Swift, A. Tucker, and X. Liu, "Stochastic dynamic modeling of short gene expression time-series data," *IEEE Transactions on Nanobioscience*, vol. 7, no. 1, pp. 44–55, 2008.

[43] H. Xiong and Y. Choe, "Structural systems identification of genetic regulatory networks," *Bioinformatics*, vol. 24, no. 4, pp. 553–560, 2008.

[44] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267–288, 1996.

[45] E. J. Cands and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[46] J. D. Geeter, H. V. Brussel, and J. D. Schutter, "A smoothly constrained Kalman filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1171–1177, 1997.

[47] S. M. Kay, *Fundamentals of Statistical Signal Processing. Estimation Theory*, Prentice-Hall, New York, NY, USA, 1993.

[48] http://wiki.c2b2.columbia.edu/dream/.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2013, Article ID 360678, 11 pages
http://dx.doi.org/10.1155/2013/360678

Research Article

Gene Regulation, Modulation, and Their Applications in Gene Expression Data Analysis

Mario Flores,[1] Tzu-Hung Hsiao,[2] Yu-Chiao Chiu,[3] Eric Y. Chuang,[3] Yufei Huang,[1] and Yidong Chen[2,4]

[1] *Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249, USA*
[2] *Greehey Children's Cancer Research Institute, University of Texas Health Science Center at San Antonio, San Antonio, TX 78229, USA*
[3] *Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan*
[4] *Department of Epidemiology and Biostatistics, University of Texas Health Science Center at San Antonio, San Antonio, TX 78229, USA*

Correspondence should be addressed to Yufei Huang; yufei.huang@utsa.edu and Yidong Chen; cheny8@uthscsa.edu

Received 2 December 2012; Accepted 24 January 2013

Academic Editor: Mohamed Nounou

Common microarray and next-generation sequencing data analysis concentrate on tumor subtype classification, marker detection, and transcriptional regulation discovery during biological processes by exploring the correlated gene expression patterns and their shared functions. Genetic regulatory network (GRN) based approaches have been employed in many large studies in order to scrutinize for dysregulation and potential treatment controls. In addition to gene regulation and network construction, the concept of the network modulator that has significant systemic impact has been proposed, and detection algorithms have been developed in past years. Here we provide a unified mathematic description of these methods, followed with a brief survey of these modulator identification algorithms. As an early attempt to extend the concept to new RNA regulation mechanism, competitive endogenous RNA (ceRNA), into a modulator framework, we provide two applications to illustrate the network construction, modulation effect, and the preliminary finding from these networks. Those methods we surveyed and developed are used to dissect the regulated network under different modulators. Not limit to these, the concept of "modulation" can adapt to various biological mechanisms to discover the novel gene regulation mechanisms.

1. Introduction

With the development of microarray [1] and lately the next generation sequencing techniques [2], transcriptional profiling of biological samples, such as tumor samples [3–5] and samples from other model organisms, have been carried out in order to study sample subtypes at molecular level or transcriptional regulation during the biological processes [6–8]. While common data analysis methods employ hierarchical clustering algorithms or pattern classification to explore correlated genes and their functions, the genetic regulatory network (GRN) approaches were employed to scrutinize for dysregulation between different tumor groups or biological processes (see reviews [9–12]).

To construct the network, most of research is focused on methods based on gene expression data derived from high-throughput technologies by using metrics such as Pearson or Spearman correlation [13], mutual information [14], co-determination method [15, 16], Bayesian methods [17, 18], and probabilistic Boolean networks [19]. Recently, new transcriptional regulation via competitive endogenous RNA (ceRNAs) has been proposed [20, 21], introducing additional dimension in modeling gene regulation. This type of regulation requires the knowledge of microRNA (miRNA) binding targets [22, 23] and the hypothesis of RNA regulations via competition of miRNA binding. Common GRN construction tries to confine regulators to be transcription factor (TF) proteins, a primary transcription programming machine, which relies

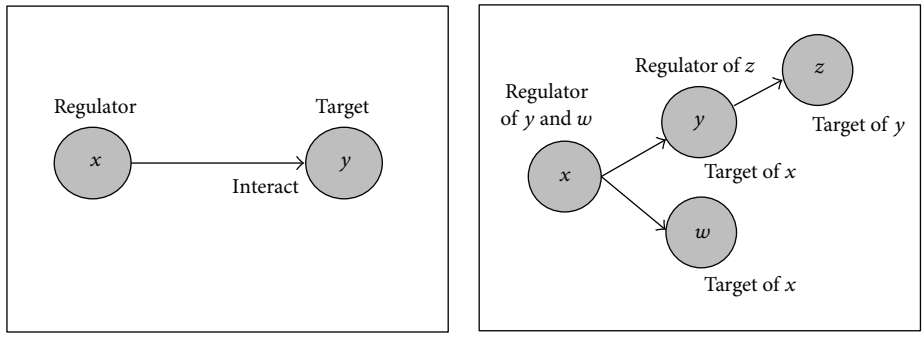


Figure 1: Regulator-target pair in genetic regulatory network model: (a) basic regulator-target pair and (b) regulator-target complex.

on sequence-specific binding sites at target genes' promoter regions. In contrast, ceRNAs mediate gene regulation via competing miRNAs binding sites in target 3′UTR region, which exist in >50% of mRNAs [22, 24]. In this study, we will extend the current network construction methods by incorporating regulation via ceRNAs.

In tumorigenesis, gene mutation is the main cause of the cancer [25]. The mutation may not directly reflect in the change at the gene expression level; however, it will disrupt gene regulation [26–28]. In Hudson et al., they found that mutated myostatin and MYL2 showed different coexpressions when comparing to wild-type myostatin. Chun et al. also showed that oncogenic KRAS modulates HIF-1α and HIF-2α target genes and in turn modulates cancer metabolism. Stelniec-Klotz et al. presented a complex hierarchical model of KRAS modulated network followed by double perturbation experiments. Shen et al. [29] showed a temporal change of GRNs modulated after the estradiol stimulation, indicating important role of estrogen in modulating GRNs. Functionally, modulation effect of high expression of *ESR1* was also reported by Wilson and Dering [30] where they studied previously published microarray data with cells treated with hormone receptor agonists and antagonists [31–33]. In this study, a comprehensive review of existing algorithms to uncover the modulators was provided. Given either mutation or protein expression status was unknown under many of reported studies, the problem of how to partition the diverse samples with different conditions, such as active or inactive oncogene status (and perhaps a combination of multiple mutations), and the prediction of a putative modulator of gene regulation remains a difficult task.

By combining gene regulation obtained from coexpression data and ceRNAs, we report here an early attempt to unify two systems mathematically while assuming a known modulator, estrogen receptor (ER). By employing the TCGA [3] breast tumor gene expressions data and their clinical test result (ER status), we demonstrate the approach of obtaining GRN via ceRNAs and a new presentation of ER modulation effects. By integrating breast cancer data into our unique ceRNAs discovery website, we are uniquely positioned to further explore the ceRNA regulation network and further develop the discovery algorithms in order to detect potential modulators of regulatory interactions.

2. Models of Gene Regulation and Modulation

2.1. Regulation of Gene Expression. The complex relationships among genes and their products in a cellular system can be studied using genetic regulatory networks (GRNs). The networks model the different states or phenotypes of a cellular system. In this model, the interactions are commonly modeled as regulator-target pairs with edges between regulator and target pair representing their interaction direction, as shown in Figure 1(a). In this model a target gene is a gene whose expression can be altered (activated or suppressed) by a regulator gene. This definition of a target gene implies that any gene can be at some point a target gene or a direct or indirect regulator depending on its position in the genetic regulatory network. The regulator gene is a gene that controls (activates or suppresses) its target genes' expression. The consequences of these activated (or suppressed) genes sometimes are involved in specific biological functions, such as cell proliferation in cancer. Examples of regulator-target pair in biology are common. For example, a target gene CDCA7 (cell division cycle-associated protein 7) is a c-Myc (regulator) responsive gene, and it is part of c-Myc-mediated transformation of lymphoblastoid cells. Furthermore, as shown in Figure 1(b), a regulator gene can also act as a target gene if there exists an upstream regulator.

If the interaction is modeled after Boolean network (BN) model [34], then

$$y_i(t+1) = f_i\left(x_{j_1}(t), \ldots, x_{j_k}(t), y_i(t)\right), \quad (1)$$

where each regulator $x_j \in \{0, 1\}$ is a binary variable, as well as it is target y_i. As described by (1), the target y_i at time $t+1$ is completely determined by the values of its regulators at time t by means of a Boolean function $f_i \in F$, where F is a collection of Boolean functions. Thus, the Boolean network $G(V, F)$ is defined as a set of nodes (genes) $V = \{x_1, x_2, \ldots, x_n\}$ and a list of functions (edges or interactions) $F = \{f_1, f_2, \ldots, f_n\}$. Similarly such relationship can be defined in the framework of Bayesian network where the

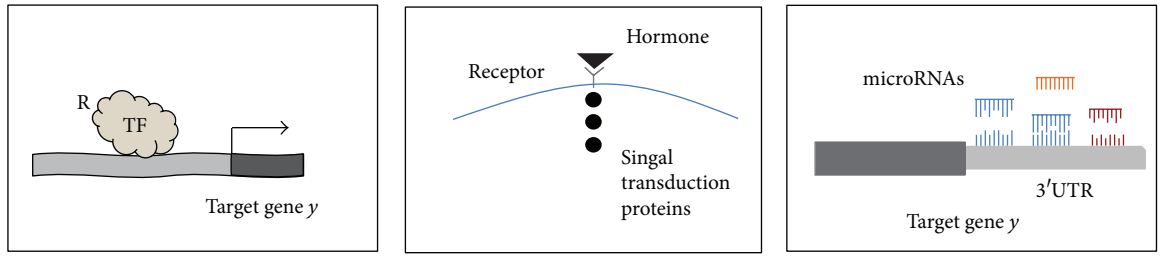


FIGURE 2: Three different cases of regulation of gene expression that share the network representation of a regulator target interaction.

similar regulators-target relationship as defined in (1) can be modeled by the distribution

$$P\left(y_i\left(t+1\right), x_{j_1}\left(t\right), \ldots, x_{j_k}\left(t\right), y_i\left(t\right)\right) = P\left(y_i\left(t+1\right) \mid \text{Parents}\left(y_i\left(t+1\right)\right)\right) \times P\left(\text{Parents}\left(y_i\left(t+1\right)\right)\right), \quad (2)$$

where $\text{Parents}(y_i(t+1)) = \{x_{j_1}(t), \ldots, x_{j_k}(t), y_i(t)\}$ is the set of regulators, or parents, of y_i, $P(y_i(t+1) \mid \text{Parents}(y_i(t+1)))$ is the conditional distribution defining the regulator-target relationship, and $P(\text{Parents}(y_i(t+1)))$ models the prior distribution of regulators. Unlike in (1), the target and regulators in (2) are modeled as random variables. Despite of this difference, in both (1) and (2), the target is always a function (or conditional distribution) of the regulator (or parents). When the relationship is defined by a Boolean function as in (1), the conditional distribution in (2) take the form of a binomial distribution (or a multinomial distribution when both regulators and target take more than two states). Other distributions such as the Gaussian and Poisson can be introduced to model more complex relationships than the Boolean. The network construction, inference, and control, however, are beyond the scope of this paper, and we leave the topics to the literatures [9, 35, 36].

The interactions among genes and their products in a complex cellular process of gene expression are diverse, governed by the central dogma of molecular biology [37]. There are different regulation mechanisms that can actuate during different stages. Figure 2 shows three different cases of regulation of gene expression. Figure 2(a) shows the case of regulation of expression in which a transcription factor (TF) regulates the expression of a protein-coding gene (in dark grey) by binding to the promoter region of target gene y. Figure 2(b) is the case of regulation at the protein level in which a ligand protein interacts with a receptor to activate relay molecules to transduce outside signals directly into cell behavior. Figure 2(c) is the case of regulation at the RNA level in which one or more miRNAs regulate target mRNA y by translational repression or target transcript degradation via binding to sequence-specific binding sites (called miRNA response elements or MREs) in 3′UTR region. As illustrated in Figure 2(c), the target genes/proteins all contain a domain of binding or docking site, enabling specific interactions between regulator-target pairs, a common element in network structure.

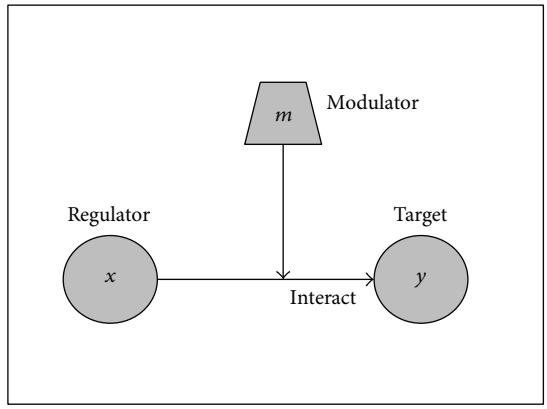


FIGURE 3: Graphical representation of the triplet interaction of regulator x, target y, and modulator m.

2.2. Modulation of Gene Regulation. Different from the concept of a coregulator commonly referred in the regulatory biology, a modulator denotes a gene or protein that is capable of altering the endogenous gene expression at one stage or time. In the context of this paper, we specifically define a modulator to be a gene that can systemically influence the interaction of regulator-target pair, either to activate or suppress the interaction in the presence/absence of the modulator. One example of modulator is the widely studied estrogen receptor (ER) in breast cancer studies [38–40]; the ER status determines not only the tumor progression, but also the chemotherapy treatment outcomes. It is well known that binding of estrogen to receptor facilitates the ER activities to activate or repress gene expression [41], thus effectively modulating the GRN. Figure 3 illustrates the model of the interaction between a modulator (m) and a regulator (x) target (y) pair that it modulates.

Following the convention used in (1) and (2), the modulation interaction in Figure 3 can be modeled by

$$y = \mathcal{F}^{(m)}\left(x\right), \quad (3)$$

where y represents target expression, x represents the parents (regulators) of target y, and $\mathcal{F}^{(m)}(\cdot)$ is the regulation function

modulated by m. When $\mathcal{F}^{(m)}(\cdot)$ is stochastic, the relationship is modeled by the conditional distribution as

$$p(y, x \mid m) = p(y \mid x, m)\, p(x \mid m), \tag{4}$$

where $p(y|x,m)$ models the regulator-target relationship modulated by m and $p(x|m)$ defines the prior distribution of regulators (parents) expression modulated by m. Different distribution models can be used to model different mechanisms for modulation. At the biological level, there are different mechanisms for modulation of the interaction x-y, and currently several algorithms for prediction of the modulators has been developed. This survey presents the latest formulations and algorithms for prediction of modulators.

3. Survey of Algorithms of Gene Regulation and Modulation Discovery

During the past years, many computational tools have been developed for regulation network construction, and then depending on the hypothesis, modulator concept can be tested and extracted. Here we will focus on modulator detection algorithms (MINDy, Mimosa, GEM, and Hermes). To introduce gene-gene interaction concept, we will also briefly discuss algorithms for regulation network construction (ARACNE) and ceRNA identification algorithm (MuTaMe).

3.1. ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks). ARACNE [14, 42] is an algorithm that extracts transcriptional networks from microarray data by using an information-theoretic method to reduce the indirect interactions. ARACNE assumes that it is sufficient to estimate 2-way marginal distributions, when sample size $M > 100$, in genomics problems, such that

$$p(x_i) = \frac{1}{z} e^{-[\sum_{i=1}^{N} \phi_i(x_i) + \sum_{i,j}^{N} \phi_{ij}(x_i, y_j)]}. \tag{5}$$

Or a candidate interaction can be identified using estimation of mutual information MI of genes x and y, $\mathrm{MI}(x, y) = \mathrm{MI}_{xy}$, where $\mathrm{MI}_{xy} = 1$ if genes x and y are identical, and MI_{xy} is zero if $p(x, y) = p(x)p(y)$, or x and y are statistically independent. Specifically, the estimation of mutual information of gene expressions x and y of regulator and target genes is done by using the Gaussian kernel estimator. The ARACNE takes additional two steps to clean the network: (1) removing MI if its P value is less than that derived from two independent genes via random permutation and (2) data processing inequality (DPI). The algorithm further assumes that for a triplet gene (g_x, g_y, g_z), where g_x regulates g_z, through g_y, then

$$\mathrm{MI}_{x,z} < \min\left(\mathrm{MI}_{x,y}, \mathrm{MI}_{y,z}\right), \quad \text{if } x \longrightarrow y \longrightarrow z, \text{ with no alternative path}, \tag{6}$$

where $\rightarrow$ represents regulation relationship. In other words, the lowest mutual information $\mathrm{MI}_{x,z}$ is from an indirect interaction and thus shall be removed from the GRN by ARACNE in the DPI step. A similar algorithm was proposed [43] to utilize conditional mutual information to explore more than 2 regulators.

3.2. MINDy (Modulator Inference by Network Dynamics). Similar to ARACNE, MINDy is also an information-theoretic algorithm [44]. However, MINDy aims to identify potential transcription factor-(TF-target) gene pairs that can be modulated by a candidate modulator. MINDy assumes that the expressions of the modulated TF-target pairs are of different correlations under different expression state of the modulator. For simplicity and computational consideration, MINDy considers only two modulator expression states, that is, up- ($m = 1$) or down-expression ($m = 0$). Then, it tests if the expression correlations of potential TF-target pairs are significantly different for modulator up-expression versus down-expression. The modulator dependent correlation is assessed by the conditional mutual information (CMI) or $I(x, y \mid m = 0)$ and $I(x, y \mid m = 1)$. Similar to ARACNE, the CMI is calculated using the Gaussian kernel estimator. To test if a pair of TF (y) and target (x) is modulated by m, the CMI difference can be calculated as

$$\Delta I = I(x, y \mid m = 1) - I(x, y \mid m = 0). \tag{7}$$

The pair is determined to be modulated if $\Delta I \neq 0$. The significance P values for $\Delta I \neq 0$ is computed using permutation tests.

3.3. Mimosa. Similarly to MINDy, Mimosa [45] was proposed to identify modulated TF-target pairs. However, it does not preselect a set of modulators of interest but rather aims to also search for the modulators. Mimosa also assumes that a modulator takes only two states, that is, absence and presence or 0 and 1. The modulated regulator-target pair is further assumed to be correlated when a modulator is present but uncorrelated when it is absent. Therefore, the distribution of a modulated TF-target pair, x and y, naturally follows a mixture distribution

$$p(x, y) = \pi p(x, y \mid m = 0) + (1 - \pi)\, p(x, ym = 1), \tag{8}$$

where π is the probability of the modulator being absent. Particularly, an uncorrelated and correlated bivariant Gaussian distributions were introduced to model different modulated regulator-target relationship, such that

$$p(x, y \mid m = 0) = \frac{1}{2\pi} e^{-(1/2)(x^2 + y^2)}, \tag{9a}$$

$$p(x, y \mid m = 1) = \frac{1}{2\pi\sqrt{1 - \alpha^2}} e^{-(1/2)(x^2 + y^2 + 2\alpha xy)/(1 - \alpha^2)}, \tag{9b}$$

where α models the correlation between x and y when the modulator is present. With this model, Mimosa sets out to fit the samples of every pair of potential regulator target with the mixture model (7). This is equivalent to finding

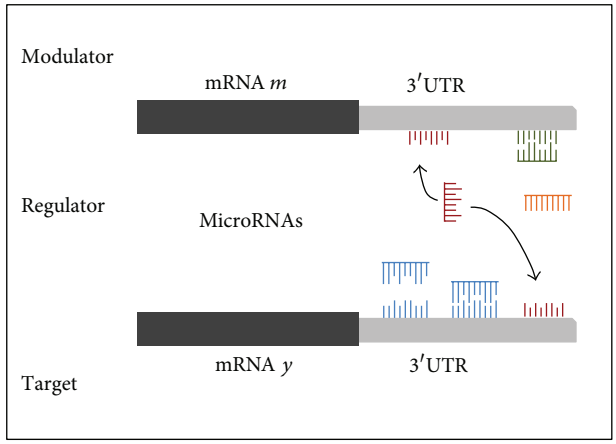


Figure 4: Modulation of gene regulation by competing mRNAs.

a partition of the paired expression samples into the correlated and uncorrelated samples. The paired expression samples that possess such correlated-uncorrelated partition ($0.3 < \pi < 0.7$ and $|\alpha| > 0.8$) are determined to be modulated. To identify the modulator of a (or a group of) modulated pair(s), a weighted t-test was developed to search for the genes whose expressions are differentially expressed in the correlated partition versus the uncorrelated partition.

3.4. GEM (Gene Expression Modulator). GEM [46] improves over MINDy by predicting how a modulator-TF interaction affects the expression of the target gene. It can detect new types of interactions that result in stronger correlation but low ΔI, which therefore would be missed by MINDy. GEM hypothesizes that the correlation between the expression of a modulator m and a target x must change, as that of the TF x changes. Unlike the previous surveyed algorithms, GEM first transforms the continuous expression levels to binary states (up- (1) or down-expression (0)) and then works only with discrete expression states. To model the hypothesized relationship, the following model is proposed:

$$P(x = 1 \mid y, m) = \alpha_c + \alpha_m m + \alpha_y y + \gamma m y, \quad (10)$$

where α_c is a constant, α_m and α_y model the effect of modulator and TF on the target genes, and γ represents the effect of modulator-TF interaction on the target gene. If the modulator-TF interaction has an effect on x, then γ will be nonzero. For a given (x, y, m) triplets GEM devised an algorithm to estimate the model coefficients in (10) and a test to determine if γ is nonzero, or m is a modulator of x and y.

3.5. MuTaMe (Mutually Targeted MRE Enrichment). The goal of MuTaMe [21] is to identify ceRNA networks of a gene of interest (GoI) or mRNA that share miRNA response elements (MREs) of same miRNAs. Figure 4 shows two mRNAs, where one is the GoIy and the other is a candidate ceRNA or modulator m. In the figure, the miRNA represented in color red has MREs in both mRNA y and mRNA m; in this case the presence of mRNA m will start the competition with y for miRNA represented in color red.

The hypothesis of MuTaMe is that mRNAs that have many of the same MREs can regulate each other by competing for miRNAs binding. The input of this algorithm is a GoI, which is targeted by a group of miRNAs known to the user. Then, from a database of predicted MREs for the entire transcriptome, it is possible to obtain the binding sites and its predicted locations in the 3′UTR for all mRNAs. This data is used to generate scores for each mRNA based on several features:

(a) the number of miRNAs that an mRNA m shares with the GoI y;

(b) the density of the predicted MREs for the miRNA; it favors the cases in which more MREs are located in shorter distances;

(c) the distribution of the MREs for every miRNA; it favors situations in which the MREs tend to be evenly distributed;

(d) the number of MREs predicted to target m; it favors situations where each miRNA contains more MREs in m.

Then each candidate transcript m will be assigned a score that results from multiplying the scores in (a) to (d). This score indicates the likelihood of the candidates to be ceRNAs and will be used to predict ceRNAs.

3.6. Hermes. Hermes [20] is an extension of MINDy that infers candidate modulators of miRNA activity from expression profiles of genes and miRNAs of the same samples. Hermes makes inferences by estimating the MI and CMI. However, different from MINDy (7), Hermes extracts the dependences of this triplet by studying the difference between the CMI of x expression and y expression conditional on the expression of m and the MI of x and y expressions as follows:

$$I = I(x; y \mid m) - I(x; y). \quad (11)$$

These quantities and their associated statistical significance can be computed from collections of expression of genes with number of samples 250 or greater. Hermes expands MINDy by providing the capacity to identify candidate modulator genes of miRNAs activity. The presence of these modulators (m) will affect the relation between the expression of the miRNAs targeting a gene (x) and the expression level of this gene (x).

In summary, we surveyed some of the most popular algorithms for the inference of modulator. Additional modulator identification algorithms are summarized in Table 1. It is worth noting that the concept of modulator applies to cases beyond discussed in this paper. Such example includes the multilayer integrated regulatory model proposed in Yan et al. [49], where the top layer of regulators could be also considered as "modulators."

4. Applications to Breast Cancer Gene Expression Data

Algorithms of utilizing modulator concept have been implemented in various software packages. Here we will discuss

Table 1: Gene regulation network and modulator identification methods.

Algorithm	Features	References
ARACNE	Interaction network constructed via mutual information (MI).	[14, 42]
Network profiler	A varying-coefficient structural equation model (SEM) to represent the modulator-dependent conditional independence between genes.	[47]
MINDy	Gene-pair interaction dependency on modulator candidates by using the conditional mutual information (CMI).	[44]
Mimosa	Search for modulator by partition samples with a Gaussian mixture model.	[45]
GEM	A probabilistic method for detecting modulators of TFs that affect the expression of target gene by using a priori knowledge and gene expression profiles.	[46]
MuTaMe	Based on the hypothesis that shared MREs can regulate mRNAs by competing for microRNAs binding.	[21]
Hermes	Extension of MINDy to include microRNAs as candidate modulators by using CMI and MI from expression profiles of genes and miRNAs of the same samples.	[20]
ERα modulator	Analyzes the interaction between TF and target gene conditioned on a group of specific modulator genes via a multiple linear regression.	[48]

two new applications, MEGRA and TraceRNA, implemented in-house specifically to utilize the concept of differential correlation coefficients and ceRNAs to construct a modulated GRN with a predetermined modulator. In the case of MGERA, we chose estrogen receptor, *ESR1*, as the initial starting point, since it is one of the dominant and systemic factor in breast cancer; in the case of TraceRNA, we also chose gene *ESR1* and its modulated gene network. Preliminary results of applications to TCGA breast cancer data are reported in the following 2 sections.

4.1. MGERA. The Modulated Gene Regulation Analysis algorithm (MGERA) was designed to explore gene regulation pairs modulated by the modulator m. The regulation pairs can be identified by examining the coexpression of two genes based on Pearson correlation (similar to (7) in the context of correlation coefficient). Fisher transformation is adopted to normalize the correlation coefficients biased by sample sizes to obtain equivalent statistical power among data with different sample sizes. Statistical significance of difference in the absolute correlation coefficients between two genes is tested by the student t-test following Fisher transformation. For the gene pairs with significantly different coefficients between two genes, active and deactive statuses are identified by examining the modulated gene expression pairs (MGEPs). The MGEPs are further combined to construct the m modulated gene regulation network for a systematic and comprehensive view of interaction under modulation.

To demonstrate the ability of MGERA, we set estrogen receptor (ER) as the modulator and applied the algorithm to TCGA breast cancer expression data [3] which contains 588 expression profiles (461 ER+ and 127 ER−). By using P value <0.01 and the difference in the absolute Pearson correlation coefficients >0.6 as criteria, we identified 2,324 putative ER+ MGEPs, and a highly connected ER+ modulated gene regulation network was constructed (Figure 5). The top ten genes with highest connectivity was show in Table 2. The cysteine/tyrosine-rich 1 gene (*CYYR1*), connected to 142 genes, was identified as the top hub gene in the network and thus may serve as a key regulator under ER+ modulation.

Table 2: Hub genes derived from modulated gene regulation network (Figure 5).

Gene	Number of ER+ MGEPs
CYYR1	142
MRAS	109
C9orf19	95
LOC339524	93
PLEKHG1	92
FBLN5	91
BOC	91
ANKRD35	89
FAM107A	83
C16orf77	73

Gene Ontology analysis of *CYYR1* and its connected neighbor genes revealed significant association with extracellular matrix, epithelial tube formation, and angiogenesis.

4.2. TraceRNA. To identify the regulation network of ceRNAs for a GoI, we developed a web-based application TraceRNA presented earlier in [50] with extension to regulation network construction. The analysis flow chart of TraceRNA was shown in Figure 6. For a selected GoI, the GoI binding miRNAs (GBmiRs) were derived either validated miRNAs from miRTarBase [51] or predicted miRNAs from SVMicrO [52]. Then mRNAs (other than the given GoI) also targeted by GBmiRs were identified as the candidates of ceRNAs. The relevant (or tumor-specific) gene expression data were used to further strengthen relationship between the ceRNA candidates and GoI. The candidate ceRNAs which coexpressed with GoI were reported as putative ceRNAs. To construct the gene regulation network via GBmiRs, we set each ceRNA as the secondary GoI, and the ceRNAs of these secondary GoIs were identified by applying the algorithm recursively. Upon identifying all the ceRNAs, the regulation network of ceRNAs of a given GoI was constructed.

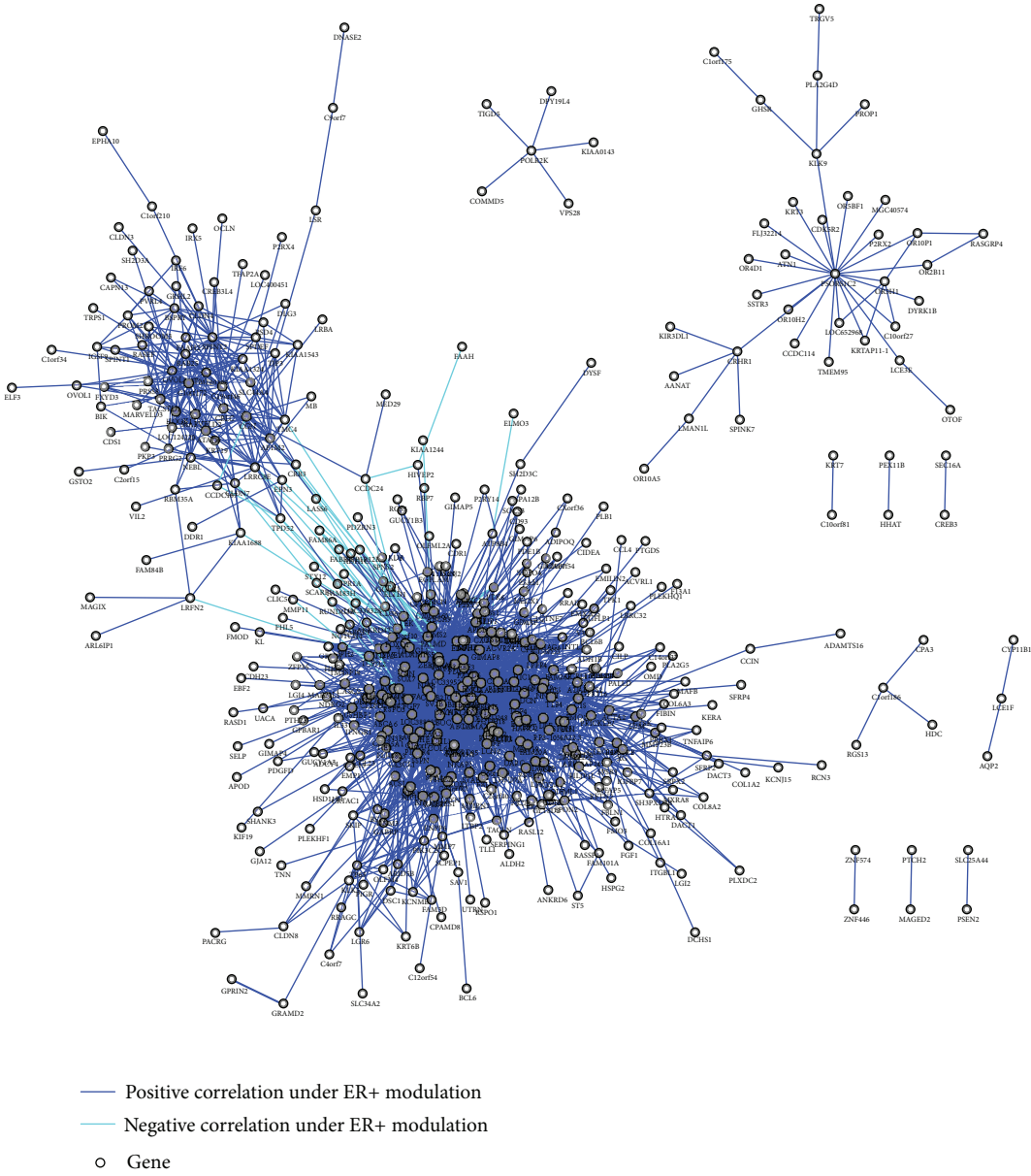


Figure 5: ER+ modulated gene regulation network.

To identify ceRNA candidates, three miRNAs binding prediction algorithms, SiteTest, SVMicrO, and BCMicrO, were used in TracRNA. SiteTest is an algorithm similar to MuTaMe and uses UTR features for target prediction. SVMicrO [52] is an algorithm that uses a large number of sequence-level site as well as UTR features including binding secondary structure, energy, and conservation, whereas BCMicrO [53] employs a Bayesian approach that integrates predictions from 6 popular algorithms including TargetScan, miRanda, PicTar, mirTarget, PITA, and DIANA-microT. Pearson correlation coefficient was used to test the coexpression between the GoI and the candidate ceRNAs. We utilized TCGA breast cancer cohort [3] as the expression data, by using 60% of GBmiRs as common miRNAs and Pearson correlation coefficient >0.9 as criteria. The final scores of putative ceRNAs (see Table 3, last column) were generated by using Borda merging method which rerank the sum of ranks from both GBmiR binding and coexpression *P* values [54]. To illustrate the utility of the TraceRNA algorithm for breast cancer study, we also focus on the genes interacted with the estrogen receptor alpha, *ESR1*, with GBmiRs including *miR-18a, miR-18b, miR-193b, miR-19a, miR-19b, miR-206, miR-20b, miR-22, miR-221, miR-222, miR-29b, and miR-302c*. The regulation network generated by *ESR1* as the initial GoI is shown in Figure 7, and the top 18 ceRNAs are provided in Table 3. The TraceRNA algorithm can be accessed http://compgenomics.utsa.edu/cerna/.

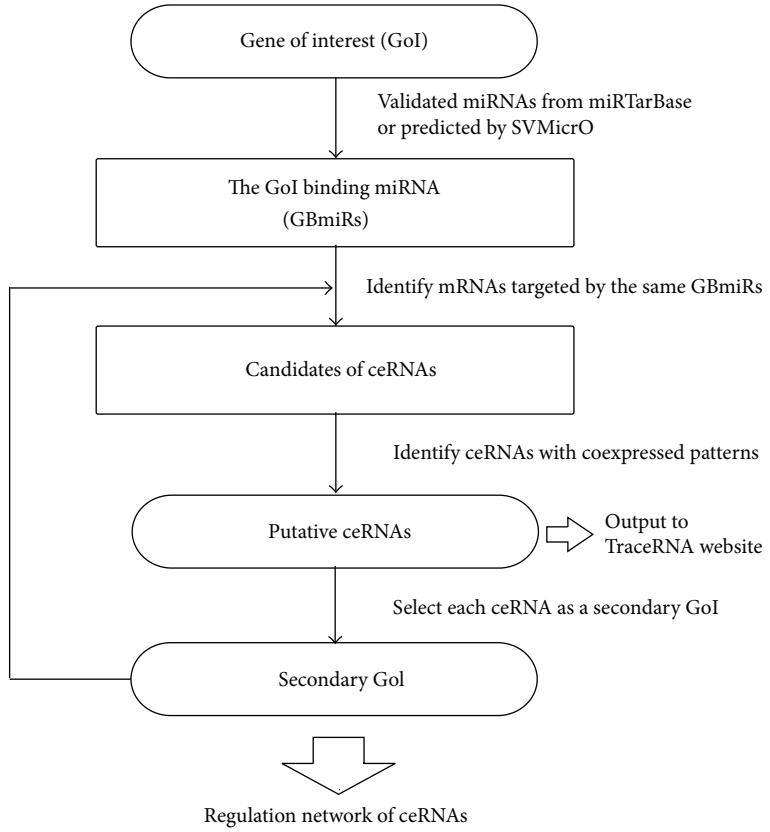
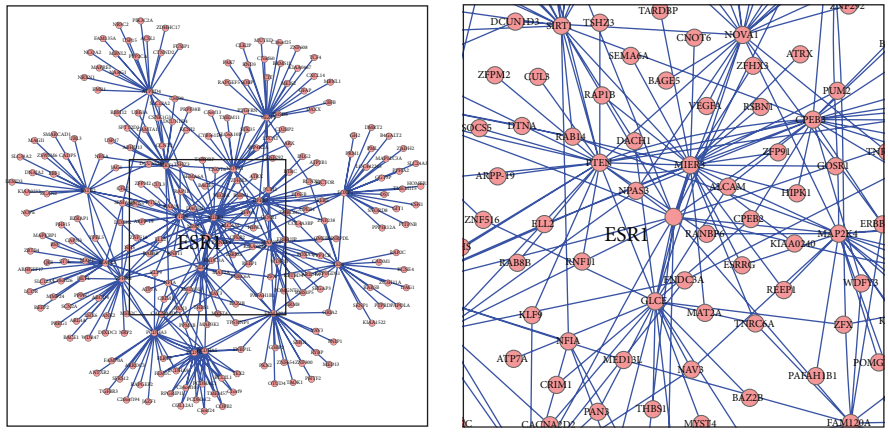


Figure 6: The analysis flow chart of TraceRNA.



(a) (b)

Figure 7: (a) ceRNA network for gene of interest ESR1 generated using TraceRNA. (b) Network graph enlarged at ESR1.

5. Conclusions

In this report, we attempt to provide a unified concept of modulation of gene regulation, encompassing earlier mRNA expression based methods and lately the ceRNA method. We expect the integration of ceRNA concept into the gene-gene interactions, and their modulator identification will further enhance our understanding in gene interaction and their systemic influence. Applications provided here also represent examples of our earlier attempt to construct modulated networks specific to breast cancer studies. Further investigation will be carried out to extend our modeling to provide a unified understanding of genetic regulation in an altered environment.

Table 3: Top 18 candidate ceRNAs for *ESR1* as GOI obtained from TraceRNA. *ESR1* is at rank of 174 (not listed in this table).

Gene symbol	SVMicrO-based prediction		Expression correlation		Final score
	Score	*P* value	Score	*P* value	
FOXP1	1.066	0.0043	0.508	0.016	1212
VEZF1	0.942	0.0060	0.4868	0.020	1179
NOVA1	0.896	0.0067	0.479	0.023	1160
CPEB3	0.858	0.0074	0.484	0.022	1149
MAP2K4	0.919	0.0064	0.322	0.097	1139
FAM120A	0.885	0.0069	0.341	0.082	1130
PCDHA3	0.983	0.0054	0.170	0.215	1125
SIRT1	0.927	0.0062	0.230	0.162	1117
PCDHA5	0.983	0.0054	0.148	0.233	1113
PTEN	0.898	0.0067	0.221	0.168	1104
PCDHA1	0.983	0.0054	0.140	0.239	1103
NBEA	0.752	0.0098	0.491	0.020	1102
ZFHX4	0.970	0.0056	0.154	0.229	1097
GLCE	0.798	0.0087	0.3231	0.096	1096
MAGI2	0.777	0.0092	0.321	0.097	1086
SATB2	0.801	0.0086	0.243	0.151	1078
LEF1	0.753	0.0098	0.291	0.112	1065
ATPBD4	0.819	0.0082	0.170	0.215	1060

Authors' Contribution

M. Flores and T.-H Hsiao are contributed equally to this work.

Acknowledgments

The authors would like to thank the funding support of this work by Qatar National Research Foundation (NPRP 09 -874-3-235) to Y. Chen and Y. Huang, National Science Foundation (CCF-1246073) to Y. Huang. The authors also thank the computational support provided by the UTSA Computational Systems Biology Core Facility (NIH RCMI 5G12RR013646-12).

References

[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] E. R. Mardis, "Next-generation DNA sequencing methods," *Annual Review of Genomics and Human Genetics*, vol. 9, pp. 387–402, 2008.

[3] Cancer Genome Atlas Network, "Comprehensive molecular portraits of human breast tumours," *Nature*, vol. 490, pp. 61–70, 2012.

[4] D. Bell, A. Berchuck, M. Birrer et al., "Integrated genomic analyses of ovarian carcinoma," *Nature*, vol. 474, no. 7353, pp. 609–615, 2011.

[5] R. McLendon, A. Friedman, D. Bigner et al., "Comprehensive genomic characterization defines human glioblastoma genes and core pathways," *Nature*, vol. 455, no. 7216, pp. 1061–1068, 2008.

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

[7] J. Lapointe, C. Li, J. P. Higgins et al., "Gene expression profiling identifies clinically relevant subtypes of prostate cancer," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 3, pp. 811–816, 2004.

[8] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 25, pp. 14863–14868, 1998.

[9] T. Schlitt and A. Brazma, "Current approaches to gene regulatory network modelling," *BMC Bioinformatics*, vol. 8, supplement 6, article S9, 2007.

[10] H. Hache, H. Lehrach, and R. Herwig, "Reverse engineering of gene regulatory networks: a comparative study," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2009, Article ID 617281, 2009.

[11] W. P. Lee and W. S. Tzou, "Computational methods for discovering gene networks from expression data," *Briefings in Bioinformatics*, vol. 10, no. 4, pp. 408–423, 2009.

[12] C. Sima, J. Hua, and S. Jung, "Inference of gene regulatory networks using time-series data: a survey," *Current Genomics*, vol. 10, no. 6, pp. 416–429, 2009.

[13] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science*, vol. 302, no. 5643, pp. 249–255, 2003.

[14] A. A. Margolin, I. Nemenman, K. Basso et al., "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," *BMC Bioinformatics*, vol. 7, supplement 1, article S7, 2006.

[15] 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.

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

[17] X. Chen, M. Chen, and K. Ning, "BNArray: an R package for constructing gene regulatory networks from microarray data by using Bayesian network," *Bioinformatics*, vol. 22, no. 23, pp. 2952–2954, 2006.

[18] A. V. Werhli, M. Grzegorczyk, and D. Husmeier, "Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks," *Bioinformatics*, vol. 22, no. 20, pp. 2523–2531, 2006.

[19] 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.

[20] P. Sumazin, X. Yang, H.-S. Chiu et al., "An extensive MicroRNA-mediated network of RNA-RNA interactions regulates established oncogenic pathways in glioblastoma," *Cell*, vol. 147, no. 2, pp. 370–381, 2011.

[21] Y. Tay, L. Kats, L. Salmena et al., "Coding-independent regulation of the tumor suppressor PTEN by competing endogenous mRNAs," *Cell*, vol. 147, no. 2, pp. 344–357, 2011.

[22] D. P. Bartel, "MicroRNAs: target recognition and regulatory functions," *Cell*, vol. 136, no. 2, pp. 215–233, 2009.

[23] D. Yue, J. Meng, M. Lu, C. L. P. Chen, M. Guo, and Y. Huang, "Understanding MicroRNA regulation: a computational perspective," *IEEE Signal Processing Magazine*, vol. 29, no. 1, Article ID 6105465, pp. 77–88, 2012.

[24] M. W. Jones-Rhoades and D. P. Bartel, "Computational identification of plant MicroRNAs and their targets, including a stress-induced miRNA," *Molecular Cell*, vol. 14, no. 6, pp. 787–799, 2004.

[25] D. Hanahan and R. A. Weinberg, "The hallmarks of cancer," *Cell*, vol. 100, no. 1, pp. 57–70, 2000.

[26] S. Y. Chun, C. Johnson, J. G. Washburn, M. R. Cruz-Correa, D. T. Dang, and L. H. Dang, "Oncogenic KRAS modulates mitochondrial metabolism in human colon cancer cells by inducing HIF-1α and HIF-2α target genes," *Molecular Cancer*, vol. 9, article 293, 2010.

[27] N. J. Hudson, A. Reverter, and B. P. Dalrymple, "A differential wiring analysis of expression data correctly identifies the gene containing the causal mutation," *PLoS Computational Biology*, vol. 5, no. 5, Article ID e1000382, 2009.

[28] I. Stelniec-Klotz, S. Legewie, O. Tchernitsa et al., "Reverse engineering a hierarchical regulatory network downstream of oncogenic KRAS," *Molecular Systems Biology*, vol. 8, Article ID 601, 2012.

[29] C. Shen, Y. Huang, Y. Liu et al., "A modulated empirical Bayes model for identifying topological and temporal estrogen receptor α regulatory networks in breast cancer," *BMC Systems Biology*, vol. 5, article 67, 2011.

[30] C. A. Wilson and J. Dering, "Recent translational research: microarray expression profiling of breast cancer. Beyond classification and prognostic markers?" *Breast Cancer Research*, vol. 6, no. 5, pp. 192–200, 2004.

[31] H. E. Cunliffe, M. Ringnér, S. Bilke et al., "The gene expression response of breast cancer to growth regulators: patterns and correlation with tumor expression profiles," *Cancer Research*, vol. 63, no. 21, pp. 7158–7166, 2003.

[32] J. Frasor, F. Stossi, J. M. Danes, B. Komm, C. R. Lyttle, and B. S. Katzenellenbogen, "Selective estrogen receptor modulators: discrimination of agonistic versus antagonistic activities by gene expression profiling in breast cancer cells," *Cancer Research*, vol. 64, no. 4, pp. 1522–1533, 2004.

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

[35] J. D. Allen, Y. Xie, M. Chen, L. Girard, and G. Xiao, "Comparing statistical methods for constructing large scale gene networks," *PLoS ONE*, vol. 7, no. 1, Article ID e29348, 2012.

[36] Y. Huang, I. M. Tienda-Luna, and Y. Wang, "Reverse engineering gene regulatory networks: a survey of statistical models," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 76–97, 2009.

[37] F. Crick, "Central dogma of molecular biology," *Nature*, vol. 227, no. 5258, pp. 561–563, 1970.

[38] A. Hamilton and M. Piccart, "The contribution of molecular markers to the prediction of response in the treatment of breast cancer: a review of the literature on HER-2, p53 and BCL-2," *Annals of Oncology*, vol. 11, no. 6, pp. 647–663, 2000.

[39] C. Sotiriou, S. Y. Neo, L. M. McShane et al., "Breast cancer classification and prognosis based on gene expression profiles from a population-based study," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 18, pp. 10393–10398, 2003.

[40] T. Sørlie, C. M. Perou, R. Tibshirani et al., "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 19, pp. 10869–10874, 2001.

[41] J. S. Carroll, C. A. Meyer, J. Song et al., "Genome-wide analysis of estrogen receptor binding sites," *Nature Genetics*, vol. 38, no. 11, pp. 1289–1297, 2006.

[42] K. Basso, A. A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera, and A. Califano, "Reverse engineering of regulatory networks in human B cells," *Nature Genetics*, vol. 37, no. 4, pp. 382–390, 2005.

[43] K. C. Liang and X. Wang, "Gene regulatory network reconstruction using conditional mutual information," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2008, Article ID 253894, 2008.

[44] K. Wang, B. C. Bisikirska, M. J. Alvarez et al., "Genome-wide identification of post-translational modulators of transcription factor activity in human B cells," *Nature Biotechnology*, vol. 27, no. 9, pp. 829–837, 2009.

[45] M. Hansen, L. Everett, L. Singh, and S. Hannenhalli, "Mimosa: mixture model of co-expression to detect modulators of regulatory interaction," *Algorithms for Molecular Biology*, vol. 5, no. 1, article 4, 2010.

[46] O. Babur, E. Demir, M. Gönen, C. Sander, and U. Dogrusoz, "Discovering modulators of gene expression," *Nucleic Acids Research*, vol. 38, no. 17, Article ID gkq287, pp. 5648–5656, 2010.

[47] T. Shimamura, S. Imoto, Y. Shimada et al., "A novel network profiling analysis reveals system changes in epithelial-mesenchymal transition," *PLoS ONE*, vol. 6, no. 6, Article ID e20804, 2011.

[48] H. Y. Wu et al., "A modulator based regulatory network for ERalpha signaling pathway," *BMC Genomics*, vol. 13, Supplement 6, article S6, 2012.

[49] K.-K. Yan, W. Hwang, J. Qian et al., "Construction and analysis of an integrated regulatory network derived from High-Throughput sequencing data," *PLoS Computational Biology*, vol. 7, no. 11, Article ID e1002190, 2011.

[50] M. Flores and Y. Huang, "TraceRNA: a web based application for ceRNAs prediction," in *Proceedings of the IEEE Genomic Signal Processing and Statistics Workshop (GENSIPS '12)*, 2012.

[51] S. D. Hsu, F. M. Lin, W. Y. Wu et al., "MiRTarBase: a database curates experimentally validated microRNA-target interactions," *Nucleic Acids Research*, vol. 39, no. 1, pp. D163–D169, 2011.

[52] H. Liu, D. Yue, Y. Chen, S. J. Gao, and Y. Huang, "Improving performance of mammalian microRNA target prediction," *BMC Bioinformatics*, vol. 11, article 476, 2010.

[53] Y. Dong et al., "A Bayesian decision fusion approach for microRNA target prediction," *BMC Genomics*, vol. 13, 2012.

[54] J. A. Asm and M. Montague, "Models for Metasearch," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 276–284, la, New Orleans, La, USA, 2001.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2013, Article ID 171530, 10 pages
http://dx.doi.org/10.1155/2013/171530

Research Article

Spectral Analysis on Time-Course Expression Data: Detecting Periodic Genes Using a Real-Valued Iterative Adaptive Approach

Kwadwo S. Agyepong,[1] Fang-Han Hsu,[1] Edward R. Dougherty,[1,2] and Erchin Serpedin[1]

[1] *Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA*
[2] *Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ 85004-2101, USA*

Correspondence should be addressed to Erchin Serpedin; serpedin@ece.tamu.edu

Received 26 October 2012; Accepted 23 January 2013

Academic Editor: Mohamed Nounou

Time-course expression profiles and methods for spectrum analysis have been applied for detecting transcriptional periodicities, which are valuable patterns to unravel genes associated with cell cycle and circadian rhythm regulation. However, most of the proposed methods suffer from restrictions and large false positives to a certain extent. Additionally, in some experiments, arbitrarily irregular sampling times as well as the presence of high noise and small sample sizes make accurate detection a challenging task. A novel scheme for detecting periodicities in time-course expression data is proposed, in which a real-valued iterative adaptive approach (RIAA), originally proposed for signal processing, is applied for periodogram estimation. The inferred spectrum is then analyzed using Fisher's hypothesis test. With a proper p-value threshold, periodic genes can be detected. A periodic signal, two nonperiodic signals, and four sampling strategies were considered in the simulations, including both bursts and drops. In addition, two yeast real datasets were applied for validation. The simulations and real data analysis reveal that RIAA can perform competitively with the existing algorithms. The advantage of RIAA is manifested when the expression data are highly irregularly sampled, and when the number of cycles covered by the sampling time points is very reduced.

1. Introduction

Patterns of periodic gene expression have been found to be associated with essential biological processes such as cell cycle and circadian rhythm [1], and the detection of periodic genes is crucial to advance our understanding of gene function, disease pathways, and, ultimately, therapeutic solutions. Using high-throughput technologies such as microarrays, gene expression profiles at discrete time points can be derived and hundreds of cell cycle regulated genes have been reported in a variety of species. For example, Spellman et al. applied cell synchronization methods and conducted time-course gene expression experiments on *Saccharomyces cerevisiae* [2]. The authors identified 800 cell cycle regulated genes using DNA microarrays. Also, Rustici et al. and Menges et al. identified 407 and about 500 cell cycle regulated genes in *Schizosaccharomyces pombe* and *Arabidopsis*, respectively [3, 4].

Signal processing in the frequency domain simplifies the analysis and an emerging number of studies have demonstrated the power of spectrum analysis in the detection of periodic genes. Considering the common issues of missing values and noise in microarray experiments, Ahdesmäki et al. proposed a robust detection method incorporating the fast Fourier transform (FFT) with a series of data preprocessing and hypothesis testing steps [5]. Two years later, the authors further proposed a modified version for expression data with unevenly spaced time intervals [6]. A Lomb-Scargle (LS) approach, originally used for finding periodicities in astrophysics, was developed for expression data with uneven sampling [7]. Yang et al. further improved the performance using a detrended fluctuation analysis [8]. It used harmonic regression in the time domain for significance evaluation. The method was termed "Lomb-Scargle periodogram and harmonic regression (LSPR)." Basically, these methods consists of two steps: transferring the signals into the frequency

(spectral) domain and then applying a significance evaluation test for the resulting peak in the spectral density.

While numerous methods have been developed for detecting periodicities in gene expression, most of these methods suffer from false positive errors and working restrictions to a certain extent, particularly when the time-course data contain limited time points. In addition, no algorithm seems available to resolve all of these challenges. Microarray as well as other high-throughput experiments, due to high manufacturing and preparation costs, have common characteristics of small sample size [9], noisy measurements [10], and arbitrary sampling strategies [11], thereby making the detection of periodicities highly challenging. Since the number and functions of cell cycle regulated genes, or periodic genes, remain greatly uncertain, advances in detection algorithms are urgently needed.

Recently, Stoica et al. developed a novel nonparametric method, termed the "real-valued iterative adaptive approach (RIAA)," specifically for spectral analysis with nonuniformly sampled data [12]. As stated by the authors, RIAA, an iteratively weighted least-squares periodogram, can provide robust spectral estimates and is most suitable for sinusoidal signals. These characteristics of RIAA inspired us to apply it to time-course gene expression data and conduct an examination on its performance. Herein, we incorporate RIAA with a Fisher's statistic to detect transcriptional periodicities. A rigorous comparison of RIAA with several aforementioned algorithms in terms of sensitivities and specificities is conducted through simulations and simulation results dealing with real data analysis are also provided.

In this study, we found that the RIAA algorithm can provide robust spectral estimates for the detection of periodic genes regardless of the sampling strategies adopted in the experiments or the nonperiodic nature of noise present in the measurement process. We show through simulations that the RIAA can outperform the existing algorithms particularly when the data are highly irregularly sampled, and when the number of cycles covered by the sampling time points is very few. These characteristics of RIAA fit perfectly the needs of time-course gene expression data analysis. This paper is organized as follows. In Section 2, we begin with an overview of RIAA. In Section 3, a scheme for detecting periodicities is proposed, and simulation models for performance evaluation and a real data analysis for validation purposes are presented. A complete investigation of the performance of RIAA and a rigorous comparison with other algorithms are provided in Section 4.

2. RIAA Algorithm

RIAA is an iterative algorithm developed for finding the least-squares periodogram with the utilization of a weighted function. The essential mathematics involved in RIAA is introduced in this section with the algorithm input being time-course expression data; for more details regarding RIAA, the readers are encouraged to check the original paper by Stoica et al. [12].

2.1. Basics. Suppose that the signals associated with the periodic gene expressions are composed of noise and sinusoidal components. Let $y_h(t_i)$, $i = 1, \ldots, n$, denote the time-course expression ratios of gene h at instances $t_1, \ldots, t_n$, respectively; $y_h(t_i)$ are real numbers; $\sum_{i=1}^{n} y_h(t_i) = 0$. The least-squares periodogram Φ_{lsp} is given by

$$\Phi_{lsp} = |\widehat{\alpha}(\omega)|^2, \tag{1}$$

where $\widehat{\alpha}(\omega)$ is the solution to the following fitting problem:

$$\widehat{\alpha}(\omega) = \arg\min_{\alpha(\omega)} \sum_{i=1}^{n} \left[y_h(t_i) - \alpha(\omega) e^{j\omega t_i} \right]^2. \tag{2}$$

Let $\alpha(\omega) = |\alpha(\omega)| e^{j\phi(\omega)} = \beta e^{j\theta}$, where $\beta = |\alpha(\omega)| \geq 0$ and $\theta = \phi(\omega) \in [0, 2\pi]$ refer to the amplitude and phase of $\alpha(\omega)$, respectively. The criterion in (2) can then be rewritten as

$$\sum_{i=1}^{n} \left[y_h(t_i) - \beta \cos(\omega t_i + \theta) \right]^2 + \beta^2 \sum_{i=1}^{n} \sin^2(\omega t_i + \theta). \tag{3}$$

The second term in the above equation is data independent and can be omitted from the minimization operation. Hence, the criterion (2) is simplified to

$$\left(\widehat{\beta}, \widehat{\theta}\right) = \arg\min_{\beta,\theta} \sum_{i=1}^{n} \left[y_h(t_i) - \beta \cos(\omega t_i + \theta) \right]^2. \tag{4}$$

We further apply $a = \beta \cos(\theta)$ and $b = -\beta \sin(\theta)$ and derive an equivalent of (4) as follows:

$$\left(\widehat{a}, \widehat{b}\right) = \arg\min_{a,b} \sum_{i=1}^{n} \left[y_h(t_i) - a \cos(\omega t_i) - b \sin(\omega t_i) \right]^2. \tag{5}$$

The target of interest to the fitting problem now becomes $\widehat{a}$ and $\widehat{b}$ (instead of $\alpha(\omega)$), and the solution is well known to be

$$\begin{bmatrix} \widehat{a} \\ \widehat{b} \end{bmatrix} = \mathbf{R}^{-1}\mathbf{r}, \tag{6}$$

where

$$\mathbf{R} = \sum_{i=1}^{n} \begin{bmatrix} \cos(\omega t_i)^2 & \cos(\omega t_i)\sin(\omega t_i) \\ \sin(\omega t_i)\cos(\omega t_i) & \sin(\omega t_i)^2 \end{bmatrix}, \quad \mathbf{r} = \sum_{i=1}^{n} \begin{bmatrix} \cos(\omega t_i) \\ \sin(\omega t_i) \end{bmatrix} y_h(t_i). \tag{7}$$

After $\widehat{a}$ and $\widehat{b}$ are estimated, the least-squares periodogram can be derived.

2.2. Observation Interval and Resolution. Prior to implementation of RIAA for periodogram estimation, the observation interval $[0, \omega_{\max}]$ and the resolution in terms of grid size have to be selected. To this end, the maximum frequency $\omega_{\max}$ in the observation interval without aliasing errors for sampling instances $t_1, \ldots, t_n$, can be evaluated by

$$\omega_{\max} = \frac{\omega_0}{2}, \tag{8}$$

where ω_0 is given by

$$\omega_0 = \frac{2(n-1)\pi}{\sum_{i=1}^{n-1}(t_{i+1}-t_i)}. \quad (9)$$

The observation interval $[0, \omega_{\max}]$ is hence chosen after $\omega_{\max}$ is obtained.

To ensure that the smallest frequency separation in time-course expression data with regular or irregular sampling can be adequately detected, the grid size $\Delta\omega$ is chosen to be

$$\Delta\omega = \frac{2\pi}{t_n - t_1}, \quad (10)$$

which, in fact, is the resolution limit of the least-squares periodogram. As a result, the frequency grids ω_g considered in periodogram are

$$\omega_g = g\Delta\omega, \quad g = 1, \ldots, G, \quad (11)$$

where the number of grids G is given by

$$G = \left\lfloor \frac{\omega_{\max}}{\Delta\omega} \right\rfloor. \quad (12)$$

2.3. Implementation. The following notations are introduced for the implementation of RIAA at a specific frequency ω_g:

$$\begin{aligned} \mathbf{Y} &= \left[y_h(t_1) \quad \cdots \quad y_h(t_n) \right]^T, \\ \rho_g &= \left[a(\omega_g) \quad b(\omega_g) \right]^T, \\ \mathbf{A}_g &= \left[\mathbf{c}_g \quad \mathbf{s}_g \right], \end{aligned} \quad (13)$$

where

$$\begin{aligned} \mathbf{c}_g &= \left[\cos(\omega_g t_1) \quad \cdots \quad \cos(\omega_g t_n) \right]^T, \\ \mathbf{s}_g &= \left[\sin(\omega_g t_1) \quad \cdots \quad \sin(\omega_g t_n) \right]^T, \end{aligned} \quad (14)$$

and $a(\omega_g)$ and $b(\omega_g)$ denote variables a and b at frequency ω_g, respectively.

RIAA's salient feature is the addition of a weighted matrix $\mathbf{Q}_g$ to the least-squares fitting criterion. The weighted matrix $\mathbf{Q}_g$ can be viewed as a covariance matrix encapsulating the contributions of noise and other sinusoidal components in $\mathbf{Y}$ other than ω_g to the spectrum; it is defined as

$$\mathbf{Q}_g = \mathbf{\Sigma} + \sum_{m=1, m\neq g}^{G} \mathbf{A}_m \mathbf{D}_m \mathbf{A}_m^T, \quad (15)$$

where

$$\mathbf{D}_m = \frac{a^2(\omega_g) + b^2(\omega_g)}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (16)$$

and $\mathbf{\Sigma}$ denotes the covariance matrix of noise in expression data $\mathbf{Y}$, given by

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma^2 \end{bmatrix}. \quad (17)$$

Assuming that $\mathbf{Q}_g$ is invertible, in RIAA, a weighted least-squares fitting problem is formulated and considered for finding $\widehat{a}$ and $\widehat{b}$ (instead of using (5)), and it is written in the form of matrices using (13) as follows:

$$\widehat{\rho}_g = \arg\min_{\rho_g} \left[\mathbf{Y} - \mathbf{A}_g \rho_g \right]^T \mathbf{Q}_g^{-1} \left[\mathbf{Y} - \mathbf{A}_g \rho_g \right]. \quad (18)$$

In Stoica et al. [12], the solution to (18) has been shown to be

$$\widehat{\rho}_g = \frac{\mathbf{A}_g^T \mathbf{Q}_g^{-1} \mathbf{Y}}{\mathbf{A}_g^T \mathbf{Q}_g^{-1} \mathbf{A}_g}, \quad (19)$$

and the RIAA periodogram at $\omega = \omega_g$ can be derived by

$$\Phi_{\text{riaa}}(\omega_g) = \frac{1}{n} \widehat{\rho}_g^T \left(\mathbf{A}_g^T \mathbf{A}_g \right) \widehat{\rho}_g. \quad (20)$$

From (15) and (19), it is obvious that $\mathbf{Q}_g$ and $\widehat{\rho}_g$ are dependent on each other. An iterative approach (i.e., RIAA) is hence a feasible solution to get the estimate $\widehat{\rho}_g$ and the weighted matrix $\mathbf{Q}_g$.

The iteration for estimating spectrum starts with initial estimates $\widehat{\rho}_g^0$, in which the elements $\widehat{a}$ and $\widehat{b}$ are given by (6) with $\omega = \omega_g$, $g = 1, \ldots, G$. After initialization, the first iteration begins. First, the elements $\widehat{a}$ and $\widehat{b}$ of $\widehat{\rho}_g^0$ are applied to obtain $\widehat{\mathbf{D}}_m^1$ using (16). Secondly, to get a good estimate of $\widehat{\sigma}^1$, the frequency ω_p at which the largest value-p is located in the temporary periodogram $\Phi^0(\omega_g)$, $g = 1, \ldots, G$, derived using (20) with $\widehat{\rho}_g = \widehat{\rho}_g^0$, is applied for obtaining a reversed engineered signal $\widehat{\mathbf{Y}}^0$. The elements $\widehat{y}_h(t_i)$, $i = 1, \ldots, n$, in $\widehat{\mathbf{Y}}^0$ are given by

$$\widehat{y}_h(t_i) = \sqrt{2P} \cos(\omega_p t_i + s), \quad i = 1, \ldots, n. \quad (21)$$

The phase of the cosine function s is unknown; however, $\widehat{\sigma}^1$ is estimable using

$$\widehat{\sigma}^1 = \min_{s\in[0,2\pi]} \frac{\left\| \mathbf{Y} - \widehat{\mathbf{Y}}^0 \right\|^2}{n}, \quad (22)$$

where $\|\cdot\|$ is the Euclidean norm. With estimates $\widehat{\mathbf{D}}_m^1$ and $\widehat{\sigma}^1$, the estimates $\widehat{\mathbf{Q}}_g^1$, $g = 1, \ldots, G$, in the first iteration are hence given by (15). After this, $\widehat{\mathbf{Q}}_g^1$ are inserted into the right-hand side of (19) and updated estimates $\widehat{\rho}_g^1$, $g = 1, \ldots, G$, are derived. The algorithm consists of repeating these steps and updating $\widehat{\mathbf{Q}}_g^k$ and $\widehat{\rho}_g^k$ iteratively, where k denotes the number of iterations, until a termination criterion is reached. If the process stops at the Kth iteration, then the final RIAA periodogram is given by (20) using $\widehat{\rho}_g^K$. The pseudocode in Algorithm 1 represents a concise description of the iterative RIAA process.

3. Methods

Figure 1 demonstrates our scheme for periodicity detection and algorithm comparison. The first step involves a periodogram estimation, which converts the time-course gene

Algorithm RIAA

Initialization
Use (6) to obtain the initial estimates $\hat{a}$ and $\hat{b}$ in $\hat{\rho}_g^0$.
The First Iteration
Obtain $\widehat{\mathbf{D}}_m^1$ using (16) with parameters $\hat{a}$ and $\hat{b}$ given by $\hat{\rho}_g^0$. Obtain $\hat{\sigma}^1$ using (22). Using $\widehat{\mathbf{D}}_m^1$ and $\hat{\sigma}^1$ to drive the first weighted matrix $\widehat{\mathbf{Q}}_g^1$ by (15). Update estimate $\hat{\rho}_g^1$ by (19) with $\mathbf{Q}_g = \widehat{\mathbf{Q}}_g^1$.
Updating Iteration
At the kth iteration, $k = 1, 2, \ldots$, estimates $\widehat{\mathbf{Q}}_g^k$ and $\hat{\rho}_g^k$ are iteratively updated in the same way as the first iteration.
Termination
Terminate simply after 15 iterations ($K = 15$), or when the total changes in $d_g^k = ||\hat{\rho}_g^k||$ for $g = 1, \ldots, G$, is extremely small, say, $\sqrt{\sum_{g=1}^{G}(d_g^k - d_g^{k-1})^2} < 0.005\sqrt{\sum_{g=1}^{G}(d_g^{k-1})^2}$, then $K = k$.

Algorithm 1: The pseudocode of the iterative process in RIAA.

expression ratios into the frequency domain. Three methods are considered for comparison: RIAA, LS, and a detrend LS (termed DLS), which uses an additional detrend function (developed in LSPR) before regular LS periodogram estimation is applied. The derived spectra are then analyzed using hypothesis testing. This study is conducted using a Fisher's test, with the null hypothesis that there are no periodic signals in the time domain and hence no significantly large peak in the derived spectra. The algorithm performance is evaluated and compared via simulations and receiver operating characteristic (ROC) curves. In real microarray data analysis, three published benchmark sets are utilized as standards of cell cycle genes for performance comparison.

3.1. Fisher's Test. After the spectrum of time-course expression data is obtained via periodogram estimation, a Fisher's statistic f for gene h with the null hypothesis H_0 that the peak of the spectral density is insignificant against the alternative hypothesis H_1 that the peak of the spectral density is significant is applied as

$$f_h = \frac{\max_{1\le g\le G}\left(\Phi\left(\omega_g\right)\right)}{G^{-1}\sum_{g=1}^{G}\Phi\left(\omega_g\right)}, \tag{23}$$

where Φ refers to the periodogram derived using RIAA, LS, or DLS. The null hypothesis H_0 is rejected, and the gene h is claimed as a periodic gene if its p-value, denoted as p_h, is less than or equal to a specific significance threshold. For simplicity, p_h is approximated from the asymptotic null distribution of f assuming Gaussian noise [13] as follows:

$$p_h = 1 - e^{-ne^{-f_h}}. \tag{24}$$

In real data analysis, deviation might be invoked for the estimation of p_h when the time-course data is short. This issue was carefully addressed by Liew et al. [14], and, as suggested, alternative methods such as random permutation may provide less deviation and better performance. However, permutation also has limitations such as tending to be conservative [15]. While finding the most robust method for the p-value evaluation remains an open question, it gets beyond the scope of this study since the algorithm comparison via ROC curves is threshold independent [16], and the results are unaffected by the deviation.

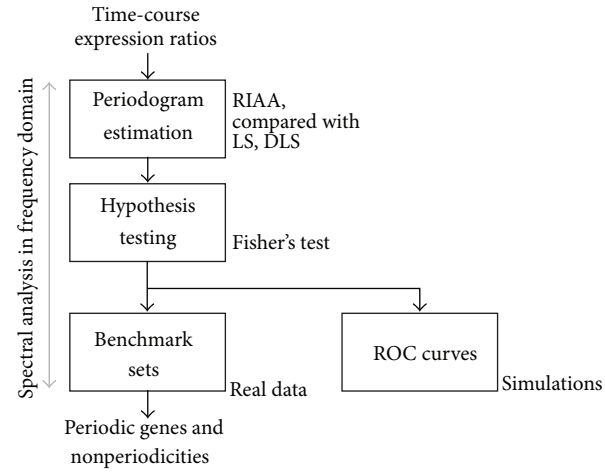


Figure 1: The scheme of the process for detecting periodicities in time-course expression data.

3.2. Simulations. Simulations are applied to evaluate the performance of RIAA. The simulation models and sampling strategies used for simulations are described in the following paragraphs.

3.2.1. Periodic and Nonperiodic Signals. Three models, one for periodic signals and two for nonperiodic signals, are considered as transcriptional signals. Since periodic genes are transcribed in an oscillatory manner, the expression levels y_s embedded with periodicities are assumed to be

$$y_s\left(t_i\right) = M\cos\left(\omega_s t_i\right) + \epsilon_{t_i}, \quad i = 1, \ldots, n, \tag{25}$$

where M denotes the sinusoidal amplitude; ω_s refers to the signal frequency; ϵ_{t_i} are Gaussian noise independent and

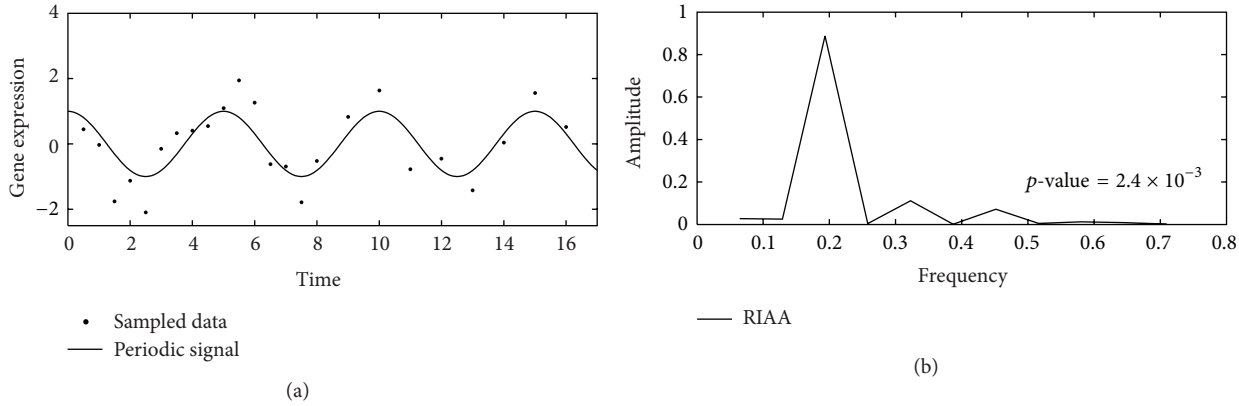


Figure 2: (a) A time-course periodic signal with frequency = 0.2 sampled by the bio-like sampling strategy; 16 time points are assigned to the interval (0,8], and 8 time points are assigned to the interval (8,16]. (b) The periodogram derived using RIAA. The maximum value (peak) in the periodogram locates at frequency = 0.195.

identically distributed (i.i.d.) with parameters μ and σ. For nonperiodic signals, the first model y_n is simply composed of Gaussian noise, given by

$$y_n(t_i) = \epsilon_{t_i}, \quad i = 1, \ldots, n. \tag{26}$$

Additionally, as visualized by Chubb et al., gene transcription can be nonperiodically activated with irregular intervals in a living eukaryotic cell, like pulses turning on and off rapidly and discontinuously [17]. Based on this, the second nonperiodic model y_n' incorporates one additional transcriptional burst and one additional sudden drop into the Gaussian noise, which can be written as

$$y_n'(t_i) = I_b(t_i) - I_d(t_i) + \epsilon_{t_i}, \quad i = 1, \ldots, n, \tag{27}$$

where I_b and I_d are indicator functions, equal to 1 at the location of the burst and the drop, respectively, and 0 otherwise. The transcriptional burst assumes a positive pulse while the transcriptional drop assumes a negative pulse. Both of them may be located randomly among all time points and are assumed to last for two time points. In other words, the indicator functions are equal to 1 at two consecutive time points, say, $I_b = 1$ at t_i and t_{i+1}. The burst and the drop have no overlap.

3.2.2. Sampling Strategies. As for the choices of sampling time points t_i, $i = 1, \ldots, n$, four different sampling strategies, one with regular sampling and three with irregular sampling, are considered. First, regular sampling is applied in which all time intervals are set to be $1/c$, where c is a constant. Secondly, a bio-like sampling strategy is invoked. This strategy tends to have more time points at the beginning of time-course experiments and less time points after we set the first 2/3 time intervals as $1/c$ and set the next 1/3 time intervals as $2/c$. Third, time intervals are randomly chosen between $1/c$ and $2/c$. The last sampling strategy, in which all time intervals are exponentially distributed with parameter c, is less realistic than the others but it is helpful for us to evaluate the performance of RIAA under pathological conditions.

ROC curves are applied for performance comparison. To this end, 10,000 periodic signals were generated using (25) and 10,000 nonperiodic signals were generated using either (26) or (27). Sensitivity measures the proportion of successful detection among the 10,000 periodic signals and specificity measures the proportion of correct claims on the 10,000 nonperiodic simulation datasets. Sampling time points are decided by one of the four sampling strategies and the number of time points n is chosen arbitrarily. For all ROC curves in Section 4, $c = 2$ and $n = 24$.

3.3. Real Data Analysis. Two yeast cell cycle experiments synchronized using an alpha-factor, one conducted by Spellman et al. [2] and one conducted by Pramila et al. [18], are considered for a real data analysis. The first time-course microarray data, termed dataset alpha and downloaded from the Yeast Cell Cycle Analysis Project website (http://genome-www.stanford.edu/cellcycle/), harbors 6,178 gene expression levels and 18 sampling time points with a 7-minute interval. The second time-course data, termed dataset alpha 38, is downloaded from the online portal for Fred Hutchinson Cancer Research Center's scientific laboratories (http://labs.fhcrc.org/breeden/cellcycle/). This dataset contains 4,774 gene expression levels and 25 sampling time points with a 5-minute interval. Three benchmark sets of genes that have been utilized in Lichtenberg et al. [19] and Liew et al. [20] as standards of cell cycle genes are also applied herein for performance comparison. These benchmark sets, involving 113, 352, and 518 genes, respectively, include candidates of cycle cell regulated genes in yeast proposed by Spellman et al. [2], Johansson et al. [21], Simon et al. [22], Lee et al. [23], and Mewes et al. [24] and are accessible in a laboratory website (http://www.cbs.dtu.dk/cellcycle/).

4. Results

RIAA performed well in the conducted simulations. As shown in Figure 2(a), a periodic signal (solid line) with amplitude $M = 1$ and frequency $\omega_s = 0.4\pi$ is sampled

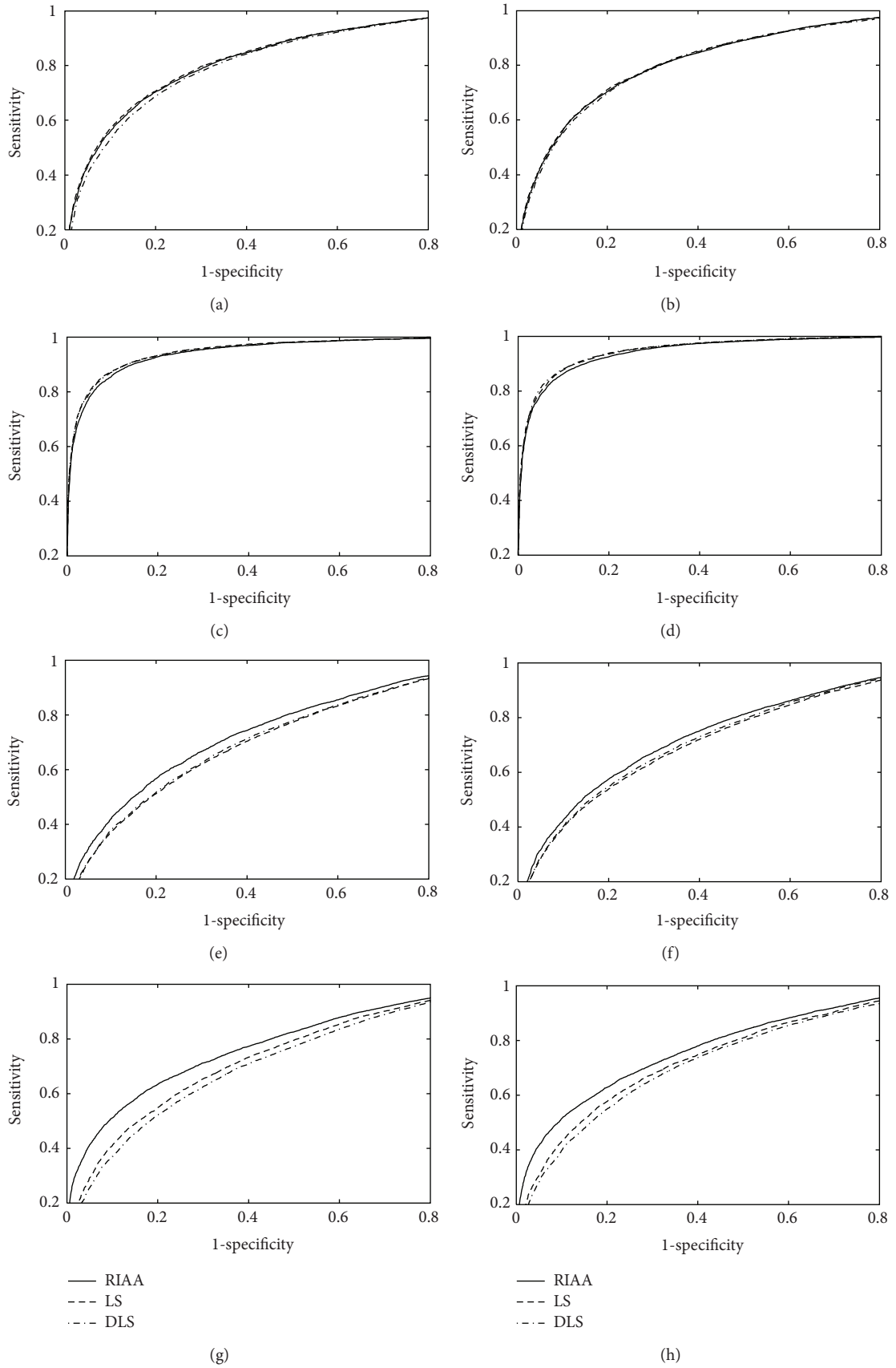


Figure 3: The ROC curves derived from simulations with 24 sampling time points, signal amplitude $M = 1$, $\omega_s = 0.4\pi$, and Gaussian noise $\mu = 0$ and $\sigma = 0.5$. Description of subplots is provided in Section 4.

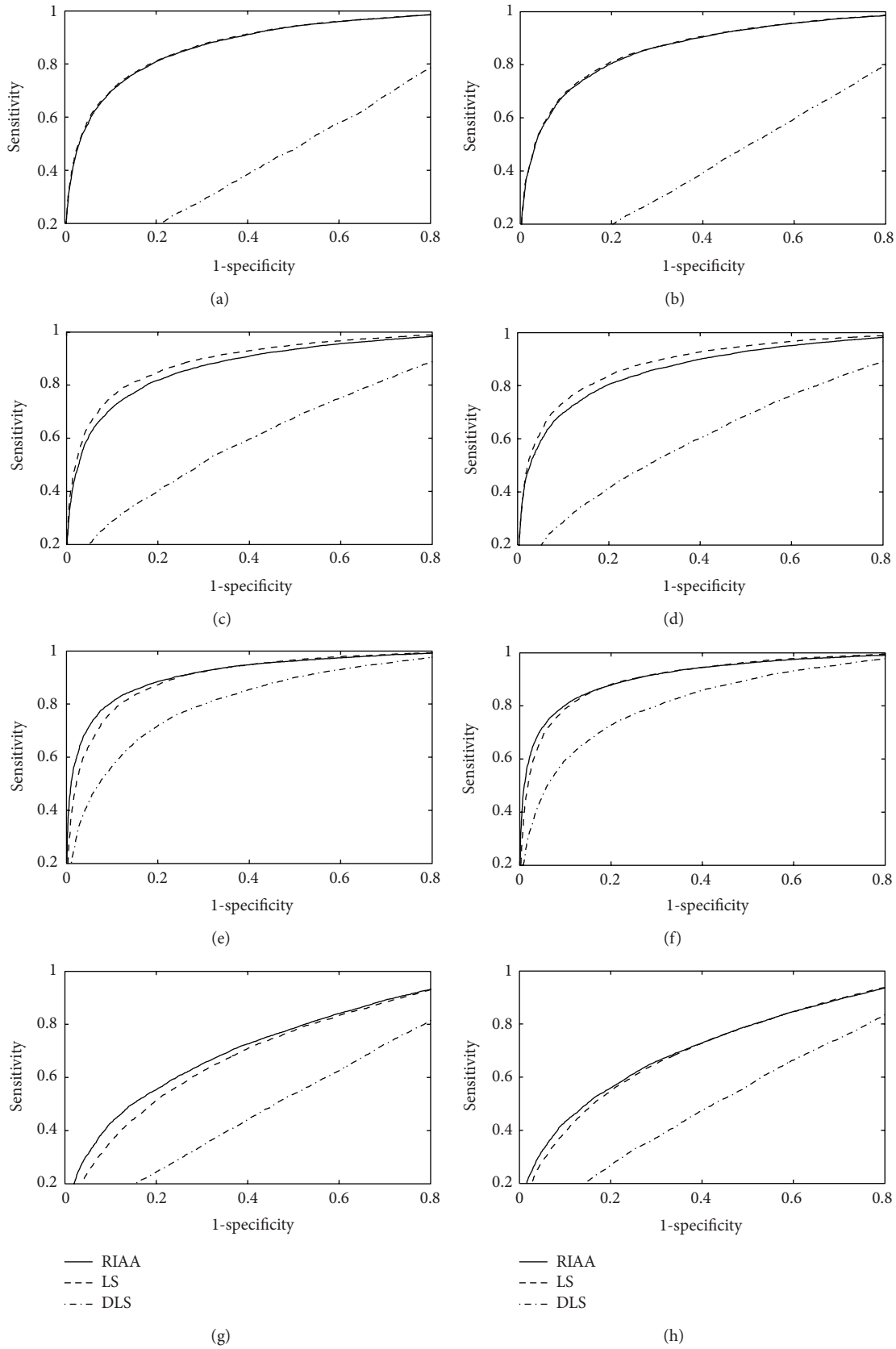


FIGURE 4: The ROC Curves derived from simulations with 24 sampling time points, signal amplitude $M = 1$, $\omega_s = 0.1\pi$, and Gaussian noise $\mu = 0$ and $\sigma = 0.5$. Description of subplots is provided in Section 4.

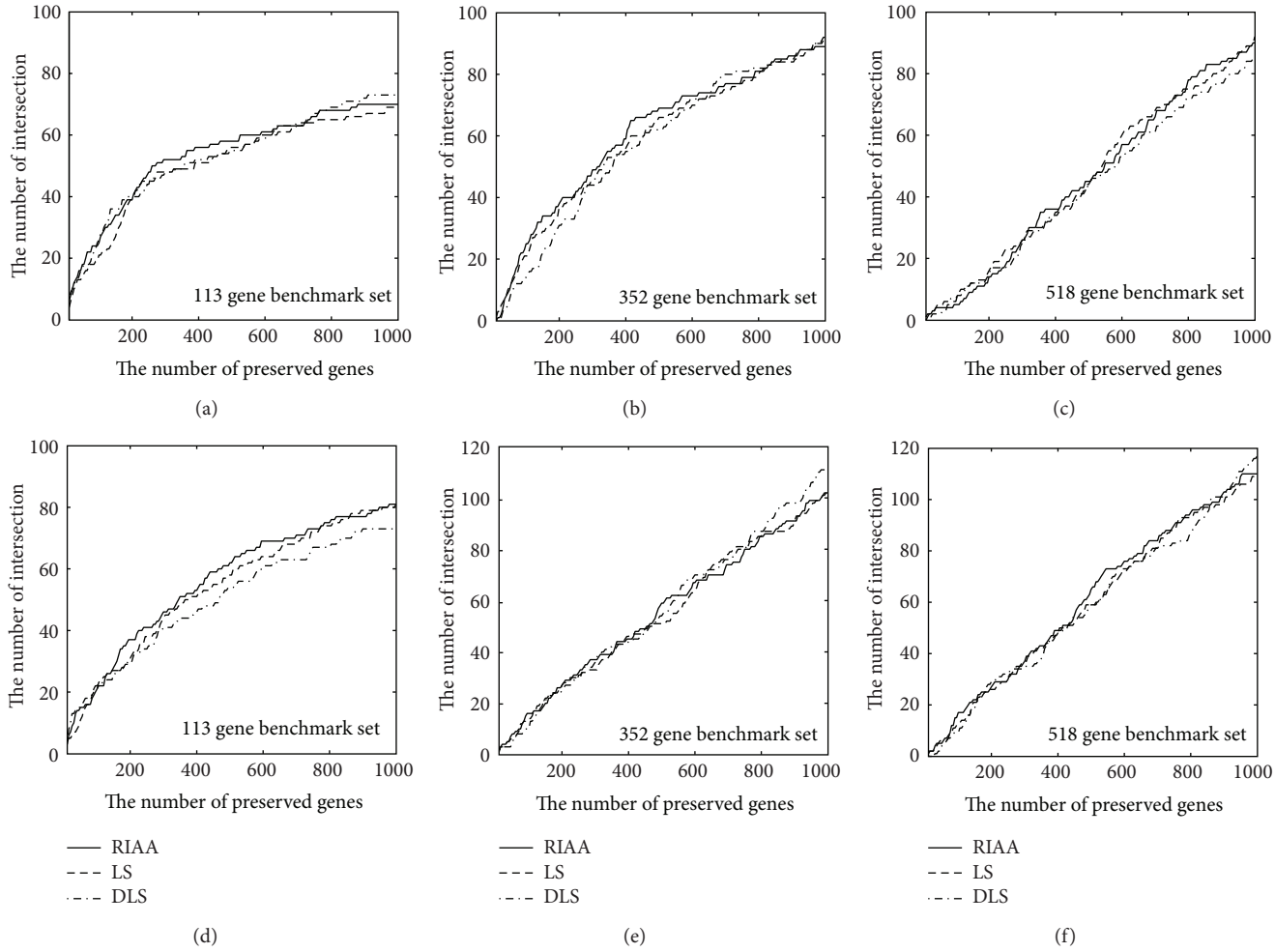


Figure 5: The intersection of preserved genes and the benchmark sets using RIAA, LS, and DLS algorithms. (a), (b), and (c) reveal the analysis results when dataset alpha was applied. (d), (e), and (f) reveal the analysis results when dataset alpha 38 was applied.

using the bio-like sampling strategy, which applies 16 time points in (0,8] and 8 more time points in (8,16]. Gaussian noise with parameters $\mu = 0$ and $\sigma = 0.5$ is assumed during microarray experiments. The resulting time-course expression levels (dots), at a total of 24 time points and the sampling time information were treated as inputs to the RIAA algorithm. Figure 2(b) demonstrates the result of periodogram estimation. In this example, the grid size $\Delta\omega$ was chosen to be 0.065 and a total of 11 amplitudes corresponding to different frequencies were obtained and shown in the spectrum. Using Fisher's test, the peak at the third grid (frequency = 0.195) was found to be significantly large (p-value = 2.4×10^{-3}), and hence a periodic gene was claimed.

ROC curves strongly illustrate the performance of RIAA. In Figures 3 and 4, subplots (a)-(b), (c)-(d), (e)-(f), and (g)-(h) refer to the simulations with regular, bio-like, binomially random, and exponentially random sampling strategies, respectively. Additionally, in the left-hand side subplots (a), (c), (e), and (g), nonperiodic signals were simply Gaussian noise with parameters $\mu = 0$ and $\sigma = 0.5$, while in the right-hand side subplots (b), (d), (f), and (h), nonperiodic signals involve not only the Gaussian noise but also a transcriptional burst and a sudden drop (27). Periodic signals were generated using (25) with amplitude $M = 1$, $c = 2$, and $n = 24$. The only difference in simulation settings between Figures 3 and 4 is the frequency of periodic signals; they are $\omega_s = 0.4\pi$ and 0.1π, respectively. As shown in these figures, LS and DLS can perform well as RIAA when the time-course data are regularly sampled, or mildly irregularly sampled; however, when data are highly irregularly sampled, RIAA outperforms the others. The superiority of RIAA over DLS is particularly clear when the signal frequency is small.

Figure 5 illustrates the results of the real data analysis when these three algorithms, namely, the RIAA, LS, and DLS, were applied. On the x-axis, the numbers indicate the thresholds η that we preserved and classified as periodicities among all yeast genes; on the y-axis, the numbers refer to the intersection of η preserved genes and the proposed periodic candidates listed in the benchmark sets. Figures 5(a)–5(c) demonstrate the results derived from dataset alpha when the 113-gene benchmark set, 352-gene benchmark

set, and 518-gene benchmark set were applied, respectively. Similarly, Figures 5(d)–5(f) demonstrate the results derived from dataset alpha 38. The RIAA does not result in significant differences in the numbers of intersections when compared to those corresponding to LS and DLS in most of these cases. However, RIAA shows slightly better coverage when the dataset alpha 38 and the 113-gene benchmark set was utilized (Figure 5(d)).

5. Conclusions

In this study, the rigorous simulations specifically designed to comfort with real experiments reveal that the RIAA can outperform the classical LS and modified DLS algorithms when the sampling time points are highly irregular, and when the number of cycles covered by sampling times is very limited. These characteristics, as also claimed in the original study by Stoica et al. [12], suggest that the RIAA can be generally applied to detect periodicities in time-course gene expression data with good potential to yield better results. A supplementary simulation further shows the superiority of RIAA over LS and DLS when multiple periodic signals are considered (see Supplementary Figure s1 available online at http://dx.doi.org/10.1155/2013/171530). From the simulations, we also learned that the addition of a transcriptional burst and a sudden drop to nonperiodic signals (the negatives) does not affect the power of RIAA in terms of periodicity detection. Moreover, the detrend function in DLS, designed to improve LS by removing the linearity in time-course data, may fail to provide improved accuracy and makes the algorithm unable to detect periodicities when transcription oscillates with a very low frequency.

The intersection of detected candidates and proposed periodic genes in the real data analysis (Figure 5) does not reveal much differences among RIAA, LS, and DLS. One possible reason is that the sampling time points conducted in the yeast experiment are not highly irregular (not many missing values are included), since, as demonstrated in Figures 3(a)–3(d), the RIAA just performs equally well as the LS and DLS algorithms when the time-course data are regularly or mildly irregularly sampled. Also, the very limited time points contained in the dataset may deviate the estimation of p-values [14] and thus hinder the RIAA from exhibiting its excellence. Besides, the number of true cell cycle genes included in the benchmark sets remains uncertain. We expect that the superiority of RIAA in real data analysis would be clearer in the future when more studies and more datasets become available.

Besides the comparison of these algorithms, it is interesting to note that the bio-like sampling strategy could lead to better detection of periodicities than the regular sampling strategy (as shown in Figures 3(c) and 3(d)). It might be beneficial to apply loose sampling time intervals at posterior periods to prolong the experimental time coverage when the number of time points is limited.

Acknowledgments

The authors would like to thank the members in the Genomic Signal Processing Laboratory, Texas A&M University, for the helpful discussions and valuable feedback. This work was supported by the National Science Foundation under Grant no. 0915444. The RIAA MATLAB code is available at http://gsp.tamu.edu/Publications/supplementary/agyepong12a/.

References

[1] W. Zhao, K. Agyepong, E. Serpedin, and E. R. Dougherty, "Detecting periodic genes from irregularly sampled gene expressions: a comparison study," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2008, Article ID 769293, 2008.

[2] 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," *Molecular Biology of the Cell*, vol. 9, no. 12, pp. 3273–3297, 1998.

[3] G. Rustici, J. Mata, K. Kivinen et al., "Periodic gene expression program of the fission yeast cell cycle," *Nature Genetics*, vol. 36, no. 8, pp. 809–817, 2004.

[4] M. Menges, L. Hennig, W. Gruissem, and J. A. H. Murray, "Cell cycle-regulated gene expression in Arabidopsis," *Journal of Biological Chemistry*, vol. 277, no. 44, pp. 41987–42002, 2002.

[5] M. Ahdesmäki, H. Lähdesmäki, R. Pearson, H. Huttunen, and O. Yli-Harja, "Robust detection of periodic time series measured from biological systems," *BMC Bioinformatics*, vol. 6, article 117, 2005.

[6] M. Ahdesmäki, H. Lähdesmäki, A. Gracey et al., "Robust regression for periodicity detection in non-uniformly sampled time-course gene expression data," *BMC Bioinformatics*, vol. 8, article 233, 2007.

[7] E. F. Glynn, J. Chen, and A. R. Mushegian, "Detecting periodic patterns in unevenly spaced gene expression time series using Lomb-Scargle periodograms," *Bioinformatics*, vol. 22, no. 3, pp. 310–316, 2006.

[8] R. Yang, C. Zhang, and Z. Su, "LSPR: an integrated periodicity detection algorithm for unevenly sampled temporal microarray data," *Bioinformatics*, vol. 27, no. 7, pp. 1023–1025, 2011.

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

[10] Y. Tu, G. Stolovitzky, and U. Klein, "Quantitative noise analysis for gene expression microarray experiments," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 22, pp. 14031–14036, 2002.

[11] Z. Bar-Joseph, "Analyzing time series gene expression data," *Bioinformatics*, vol. 20, no. 16, pp. 2493–2503, 2004.

[12] P. Stoica, J. Li, and H. He, "Spectral analysis of nonuniformly sampled data: a new approach versus the periodogram," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 843–858, 2009.

[13] J. Fan and Q. Yao, *Nonlinear Time Series: Nonparametric and Parametric Methods*, Springer, New York, NY, USA, 2003.

[14] A. W. C. Liew, N. F. Law, X. Q. Cao, and H. Yan, "Statistical power of Fisher test for the detection of short periodic gene expression profiles," *Pattern Recognition*, vol. 42, no. 4, pp. 549–556, 2009.

[15] V. Berger, "Pros and cons of permutation tests in clinical trials," *Statistics in Medicine*, vol. 19, no. 10, pp. 1319–1328, 2000.

[16] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[17] J. R. Chubb, T. Trcek, S. M. Shenoy, and R. H. Singer, "Transcriptional pulsing of a developmental gene," *Current Biology*, vol. 16, no. 10, pp. 1018–1025, 2006.

[18] T. Pramila, W. Wu, W. Noble, and L. Breeden, "Periodic genes of the yeast Saccharomyces cerevisiae: a combined analysis of five cell cycle data sets," 2007.

[19] U. Lichtenberg, L. J. Jensen, A. Fausbøll, T. S. Jensen, P. Bork, and S. Brunak, "Comparison of computational methods for the identification of cell cycle-regulated genes," *Bioinformatics*, vol. 21, no. 7, pp. 1164–1171, 2005.

[20] A. W. C. Liew, J. Xian, S. Wu, D. Smith, and H. Yan, "Spectral estimation in unevenly sampled space of periodically expressed microarray time series data," *BMC Bioinformatics*, vol. 8, article 137, 2007.

[21] D. Johansson, P. Lindgren, and A. Berglund, "A multivariate approach applied to microarray data for identification of genes with cell cycle-coupled transcription," *Bioinformatics*, vol. 19, no. 4, pp. 467–473, 2003.

[22] I. Simon, J. Barnett, N. Hannett et al., "Serial regulation of transcriptional regulators in the yeast cell cycle," *Cell*, vol. 106, no. 6, pp. 697–708, 2001.

[23] T. I. Lee, N. J. Rinaldi, F. Robert et al., "Transcriptional regulatory networks in *Saccharomyces cerevisiae*," *Science*, vol. 298, no. 5594, pp. 799–804, 2002.

[24] H. W. Mewes, D. Frishman, U. Güldener et al., "MIPS: a database for genomes and protein sequences," *Nucleic Acids Research*, vol. 30, no. 1, pp. 31–34, 2002.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2013, Article ID 618461, 8 pages
http://dx.doi.org/10.1155/2013/618461

Research Article

Identification of Robust Pathway Markers for Cancer through Rank-Based Pathway Activity Inference

Navadon Khunlertgit and Byung-Jun Yoon

Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA

Correspondence should be addressed to Byung-Jun Yoon; bjyoon@ece.tamu.edu

Received 30 November 2012; Accepted 19 January 2013

Academic Editor: Hazem Nounou

One important problem in translational genomics is the identification of reliable and reproducible markers that can be used to discriminate between different classes of a complex disease, such as cancer. The typical small sample setting makes the prediction of such markers very challenging, and various approaches have been proposed to address this problem. For example, it has been shown that pathway markers, which aggregate the gene activities in the same pathway, tend to be more robust than gene markers. Furthermore, the use of gene expression ranking has been demonstrated to be robust to batch effects and that it can lead to more interpretable results. In this paper, we propose an enhanced pathway activity inference method that uses gene ranking to predict the pathway activity in a probabilistic manner. The main focus of this work is on identifying robust pathway markers that can ultimately lead to robust classifiers with reproducible performance across datasets. Simulation results based on multiple breast cancer datasets show that the proposed inference method identifies better pathway markers that can predict breast cancer metastasis with higher accuracy. Moreover, the identified pathway markers can lead to better classifiers with more consistent classification performance across independent datasets.

1. Introduction

Advances in microarray and sequencing technologies have enabled the measurement of genome-wide expression profiles, which have spawned a large number of studies aiming to make accurate diagnosis and prognosis based on gene expression profiles [1–4]. For example, there has been significant amount of work on identifying markers and building classifiers that can be used to predict breast cancer metastasis [2, 4]. Many existing methods have directly employed gene expression data without any knowledge of the interrelations between genes. As a result, the predicted gene markers often lack interpretability and many of them are not reproducible in other independent datasets.

To overcome this problem, several different approaches have been proposed so far. For example, a recent work by Geman et al. [3] proposed an approach that utilizes the relative expression between genes, rather than their absolute expression values. It was shown that the resulting markers are easier to interpret, robust to chip-to-chip variations, and more reproducible across datasets. Another possible way to address the aforementioned problem is to interpret the gene expression data at a "modular" level through data integration [5–11]. These methods utilize additional data sources and prior knowledge—such as protein-protein interaction (PPI) data and pathway knowledge—to jointly analyze the expression of interrelated genes. This results in modular markers, such as *pathway markers* and *subnetwork markers*, which have been shown to improve the classification performance and also to be more reproducible across independent datasets [8–11]. In order to utilize pathway markers, we need to infer the pathway activity by integrating the gene expression data with pathway knowledge. For example, Guo et al. [6] used the mean or median expression value of the member genes (that belong to the same pathway) as the activity level of a given pathway. Recently, Su et al. [10] proposed a probabilistic pathway activity inference method that uses the log-likelihood ratio between different phenotypes based on the expression level of each member gene.

In this work, we propose an enhanced pathway activity inference method that utilizes the ranking of the member

genes to predict the pathway activity in a probabilistic manner. The immediate goal is to identify better pathway markers that are more reliable, more reproducible, and easier to interpret. Ultimately, we aim to utilize these markers to build accurate and robust disease classifiers. The proposed method is motivated by the relative gene expression analysis strategy proposed in [3, 12] and it builds on the concept of probabilistic pathway activity inference proposed in [10, 11]. In this study, we focus on predicting breast cancer metastasis and demonstrate that the proposed method outperforms existing methods. Preliminary results of this work have been originally presented in [13].

2. Materials and Methods

2.1. Study Datasets. Six independent breast cancer microarray gene expression datasets have been used in this study: GSE2034 (USA) [4], NKI295 (The Netherlands) [14], GSE7390 (Belgium) [15], GSE1456 (Stockholm) [16], GSE15852, and GSE9574. The Netherlands dataset uses a custom Agilent chip and it has been obtained from the Stanford website [17]. All datasets have been profiled using the Affymetrix U133a platform and they have been downloaded from the Gene Expression Omnibus (GEO) website [18].

The above datasets have been used in our study both with and without (re)normalization. To test the reproducibility of pathway markers, we selected the USA dataset and the Belgium dataset, both of which were obtained using the Affymetrix platform. The raw data for these two datasets have been normalized by utilizing the microarray preprocessing methods provided in the Bioconductor package [19]. We applied three popular normalization methods—RMA, GCRMA, and MAS5—with default setting.

The pathway data have been obtained from the MSigDB 3.0 Canonical Pathways [20]. This pathway dataset consists of 880 pathways, where 3,698 genes in these pathways intersect with all datasets.

2.2. Gene Ranking. In this study, we utilize "gene ranking" or the relative ordering of the genes based on their expression levels within each profile [3]. Consider a pathway that contains n member genes $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$ after removing the genes that are not included in all datasets. Given a sample $\mathbf{x}_k = \{x_k^1, x_k^2, \ldots, x_k^n\}$ that contains the expression level of the member genes, the gene ranking $\mathbf{r}_k$ is defined as follows:

$$\mathbf{r}_k = \left\{ r_k^{i,j} \mid 1 \le i < j \le n \right\}, \tag{1}$$

where

$$r_k^{i,j} = \begin{cases} 1, & \text{if } x_k^i < x_k^j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The resulting gene ranking $\mathbf{r}_k$ is a binary vector representing the ordering of the member genes based on their expression values in the kth sample $\mathbf{x}_k$. To preserve the gene ranking in each sample, we do not employ any between-sample normalization.

2.3. Pathway Activity Inference Based on Gene Ranking. To infer the pathway activity, we follow the strategy proposed in [10], where the activity level a_k of a given pathway in the kth sample is predicted by aggregating the probabilistic evidence of all the member genes. The main difference between the strategy proposed in this work and the original strategy [10] is that we estimate the probabilistic evidence provided by each gene based on its ranking rather than its expression value. More specifically, the pathway activity level is given by

$$a_k = \sum_{1 \le i < j \le n} \lambda_{i,j}\left(r_k^{i,j}\right), \tag{3}$$

where $\lambda_{i,j}(r_k^{i,j})$ is the log-likelihood ratio (LLR) between the two phenotypes (i.e., class labels) for the ranking $\mathbf{r}_k$. The LLR $\lambda_{i,j}(r_k^{i,j})$ is defined as

$$\lambda_{i,j}\left(r_k^{i,j}\right) = \log\left[\frac{f_{i,j}^1\left(r_k^{i,j}\right)}{f_{i,j}^2\left(r_k^{i,j}\right)}\right], \tag{4}$$

where $f_{i,j}^1(r)$ is the conditional probability mass function (PMF) of the ranking of the expression level of gene g_i and gene g_j under phenotype 1 and $f_{i,j}^2(r)$ is the conditional PMF of the ranking of the expression level of gene g_i and gene g_j under phenotype 2.

In practice, the number of possible gene pairs $\binom{n}{2}$ may be too large when we have large pathways with many member genes (i.e., when n is large). To reduce the computational complexity, we prescreen the gene pairs based on the mutual information [21] as follows. For every gene pair (i, j), we first compute the mutual information between the ranking $r_k^{i,j}$ and the corresponding phenotype c_k. Then we select the top 10% gene pairs with the highest mutual information and use only these gene pairs for computing the pathway activity level defined in (3). Although we selected the top 10% gene pairs for simplicity, this may not be necessarily optimal and one may also think of other strategies for adaptively choosing this threshold.

In a practical setting, we may not have enough training data to reliably estimate the PMFs $f_{i,j}^1(r)$ and $f_{i,j}^2(r)$. For this reason, we normalize the original LLR $\lambda_{i,j}(r_k^{i,j})$ as follows to decrease its sensitivity to small alterations in gene ranking:

$$\widehat{\lambda}_{i,j}\left(r_k^{i,j}\right) = \frac{\lambda_{i,j}\left(r_k^{i,j}\right) - \mu\left(\lambda_{i,j}\right)}{\sigma\left(\lambda_{i,j}\right)}, \tag{5}$$

where $\mu(\lambda_{i,j})$ and $\sigma(\lambda_{i,j})$ are the mean and standard deviation of $\lambda_{i,j}(r_k^{i,j})$ across all $k = 1, \ldots, n$. Figure 1 illustrates the overall process.

2.4. Assessing the Discriminative Power of Pathway Markers. In order to assess the discriminative power of a pathway marker, we compute the t-test statistics score, which is given by

$$t(\mathbf{a}) = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1/K_1 + \sigma_2/K_2}}, \tag{6}$$

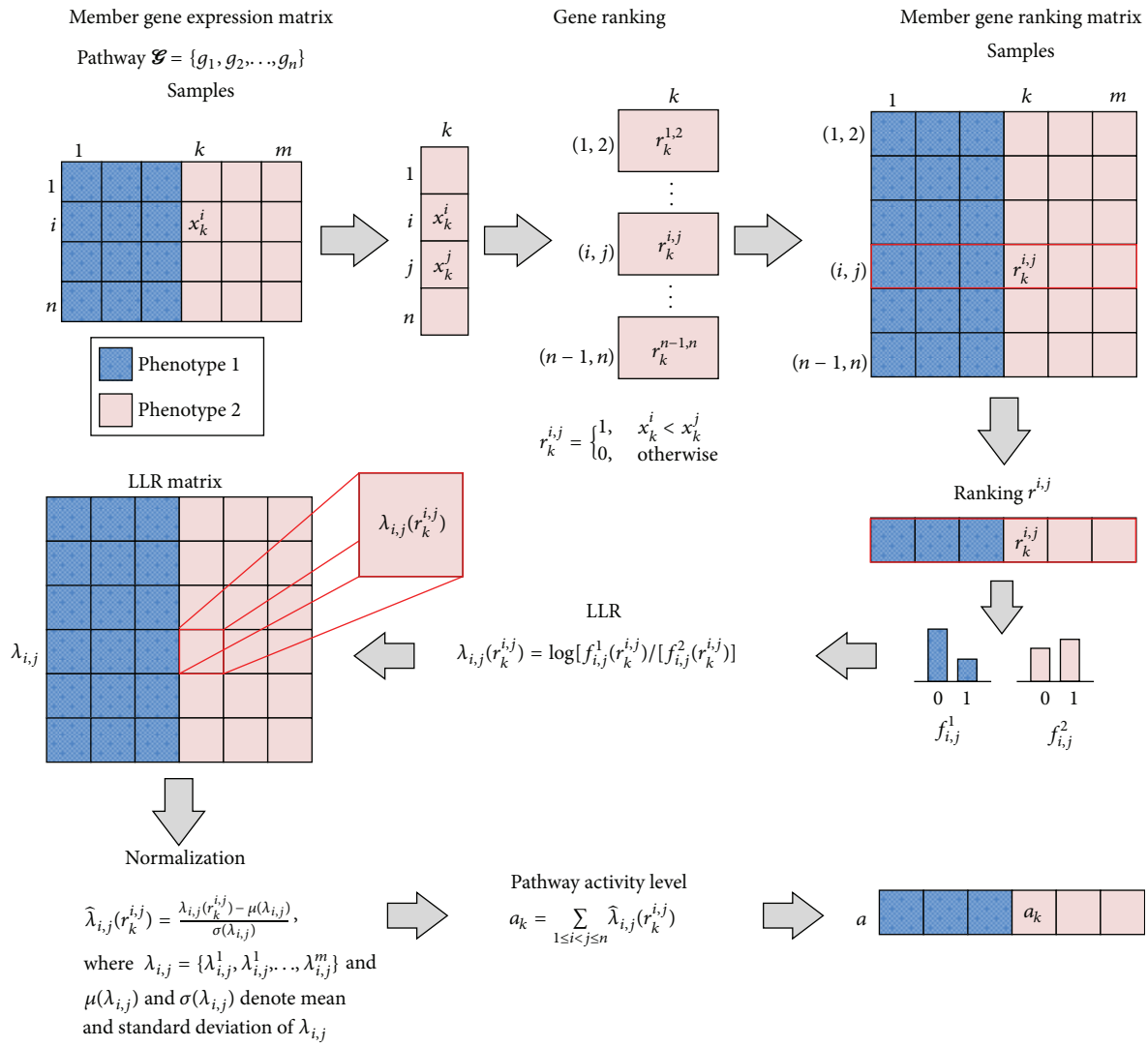


FIGURE 1: Probabilistic inference of rank-based pathway activity. For a given pathway, we first compute the ranking of the member genes for each individual sample in the dataset. Then we estimate the conditional probability mass function (PMF) of the gene ranking under each phenotype. Next, we transform the gene ranking into log-likelihood ratios (LLRs) based on the estimated PMFs and normalize the LLR matrix. Finally, the pathway activity level is inferred by aggregating the normalized LLRs of the member genes.

where $\mathbf{a} = \{a_k\}$ is the set of inferred pathway activity levels for a given pathway, μ_ℓ and σ_ℓ represent the mean and the standard deviation of the pathway activity levels for samples with phenotype $\ell \in \{1, 2\}$, respectively, and K_ℓ represents the number of samples in the dataset with phenotype ℓ. This measure has been widely used in previous studies to evaluate the performance of pathway markers [9, 10].

2.5. Evaluation of the Classification Performance. In order to evaluate the classification performance, we use the AUC (Area under ROC Curve). Many previous studies [8–11] have utilized AUC due to its ability to summarize the efficacy of a classification method over the entire range of specificity and sensitivity. We compute the AUC based on the method proposed in [22]. Given a classifier, let $x_1, x_2, \ldots, x_m$ be the output of the classifier for m positive samples and let $y_1, y_2, \ldots, y_n$ be the output for n negative samples. The AUC of the classifier can be computed as follows:

$$A = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} I\left(x_i > y_j\right), \tag{7}$$

where

$$I\left(x_i > y_j\right) = \begin{cases} 1, & \text{if } x_i > y_j, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

3. Results and Discussion

3.1. Discriminative Power of the Pathway Markers Using the Proposed Method. In order to assess the performance of the rank-based pathway activity inference method proposed in this paper, we first evaluated the discriminative power of the pathway markers following a similar setup that was adopted

in a number of previous studies [9, 10]. For comparison, we also evaluated the performance of the mean and median-based schemes proposed in [6] and the original probabilistic pathway activity inference method (we refer to this method as the "LLR method" for simplicity) presented in [10]. As explained in Materials and Methods, the discriminative power of a pathway marker was measured based on the absolute t-test score of the inferred pathway activity level. Then the pathway markers were sorted according to their t-score, in a descending order.

Figure 2 shows the discriminative power of the pathway markers on the six datasets using different activity inference methods. On each dataset, we computed the mean absolute t-test statistics score of the top P% pathways for each of the four pathway activity inference methods. The x-axis corresponds to the proportion (P%) of the top pathway markers that were considered and the y-axis shows the mean absolute t-test score for these pathway markers. As we can see from Figure 2, the proposed method clearly improves the discriminative power of the pathway markers on all six datasets that we considered in this study. In order to investigate the effect of normalization on the discriminative power of the pathway activity inference methods, we repeated this experiment using the USA and the Belgium datasets, where we first normalized the raw data using three different normalization methods (RMA, GCRMA, and MAS5) and then evaluated the discriminative power of the pathway markers. The results are summarized in Figure S1 (see Supplementary Material available online at http://dx.doi.org/10.1155/2013/618461), where we can see that the proposed rank-based scheme is not very sensitive to the choice of the normalization method and performs consistently well in all cases.

Next, we investigated how the top pathway markers identified on a specific dataset perform in other independent datasets. We first ranked the pathway markers based on their mean absolute t-test statistics score in one of the datasets and then estimated the discriminative power of the top P% markers on a different dataset. These results are shown in Figure 3, where the first dataset is used for ranking the markers and the second dataset is used for assessing the discriminative power. As we can see from Figure 3, the pathway markers identified using the mean- and the median-based schemes do not retain their discriminative power very well in other datasets. Both the LLR method [10] and the proposed rank-based inference method perform well across different datasets, where the proposed method clearly outperforms the previous LLR method. It is interesting to see that the discriminative power of the markers is retained even when we consider datasets that are obtained using different platforms. For example, USA/Belgium datasets are profiled on the U133a platform and The Netherlands dataset is profiled on a custom Agilent chip, but Figure 3 shows that pathway markers identified using the proposed method retain their discriminative power across these datasets. As before, we repeated these experiments after normalizing the datasets using different normalization methods. The results are depicted in Figure S2, where we can see that the proposed method works very well, regardless of the normalization method that was used. Interestingly, this is also true even when the first dataset and the second dataset are normalized using different methods, as shown in Figures S3 and S4.

Another interesting observation is that the rank-based method can overcome one of the limitations of the previous LLR method. For example, normalization of the Belgium dataset using GCRMA results makes the LLR method fail, as some of the genes loose variability and some of the LLR values become infinite. We can see this issue in Figures S1(d), S2(c), S3(a), and S3(f). However, this limitation is easily overcome by the proposed method through the use of gene ranking and the preselection of informative gene pairs based on mutual information.

3.2. Classification Performance of the Pathway Markers Using the Proposed Method. Next, we evaluated the classification performance of the proposed rank-based pathway activity inference method. For this purpose, we performed fivefold cross validation experiments, following a similar setup used in previous studies [8–11]. We first performed the within-dataset experiments for each of the six datasets. First, a given dataset was randomly divided into fivefolds, where fourfolds (training dataset) were used for constructing an LDA (Linear Discriminant Analysis) classifier and the remaining fold (testing dataset) was used for evaluating its performance. To construct the classifier, the training dataset was again divided into threefolds, where twofolds (marker-evaluation dataset) were used for evaluating the pathway markers and the remaining onefold (feature-selection dataset) for feature selection. The entire training dataset was used for PDF/PMF estimation. The overall setup is shown in Figure 4(a).

In order to build the classifier, we first evaluated the discriminative power of each pathway on the marker-evaluation dataset. The pathways were sorted according to their absolute t-test statistics score in a descending order and the top 50 pathways were selected as potential features. Initially, we started with an LDA-based classifier with a single feature (i.e., the pathway marker that is on the top of the list) and continued to expand the feature set by considering additional pathway markers in the list. The classifier was trained using the marker-evaluation dataset and its performance was assessed on the feature-selection dataset by measuring the AUC. Pathway markers were added to the feature set only when they increased the AUC. Finally, the performance of the classifier with the optimal feature set was evaluated by computing the AUC on the testing dataset. The above process was repeated for 100 random partitions to ensure reliable results, and we report the average AUC as the measure of overall classification performance.

Figure 5 shows how the respective classifiers that use different pathway activity inference methods perform on different datasets. As we can see in Figure 5, among the four inference methods, the proposed rank-based scheme typically yields the best average performance across these datasets. We also performed similar experiments based on the USA and the Belgium datasets after normalizing the raw data using different normalization methods. These results are summarized in Figure S5. We can see from Figure S5 that the proposed method yields the best performance on the

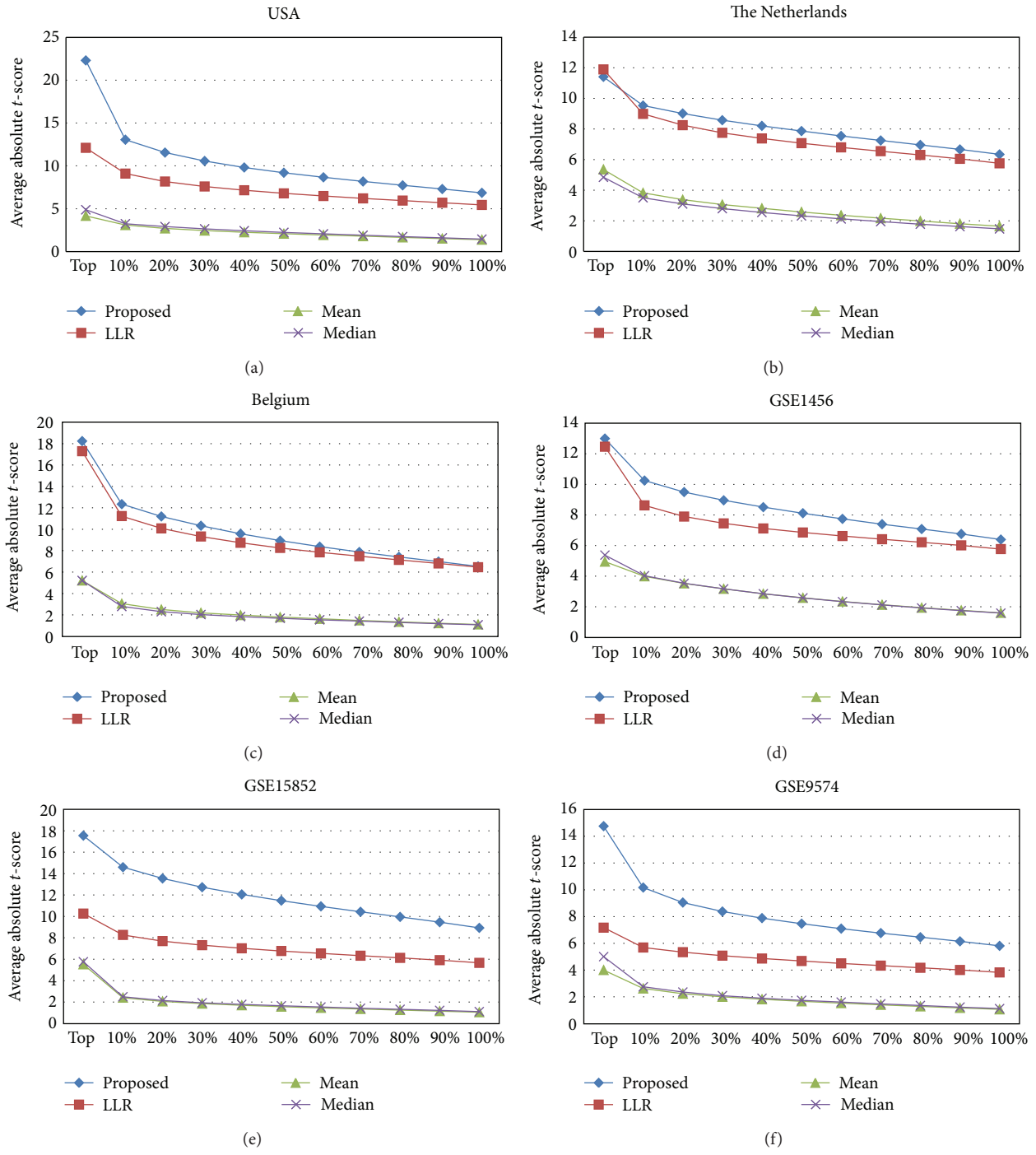


FIGURE 2: Discriminative power of pathway markers. We computed the mean absolute t-score of the top P% markers for each dataset without any further normalization.

USA dataset for all three normalization methods. On the Belgium dataset, the proposed method yields good consistent performance that is not very sensitive to the normalization method.

3.3. Reproducibility of the Pathway Markers Identified by the Proposed Method. To assess the reproducibility of the pathway markers, we performed the following cross-dataset experiments based on a similar setup that has been utilized in previous studies [8–11]. In this experiment, we used one of the breast cancer datasets for selecting the best pathway markers (i.e., only for feature selection) and a different dataset for building the classifier (using the selected pathways) and evaluating the performance of the resulting classifier. More specifically, we proceeded as follows. The first dataset was first divided into threefolds, where twofolds were used for

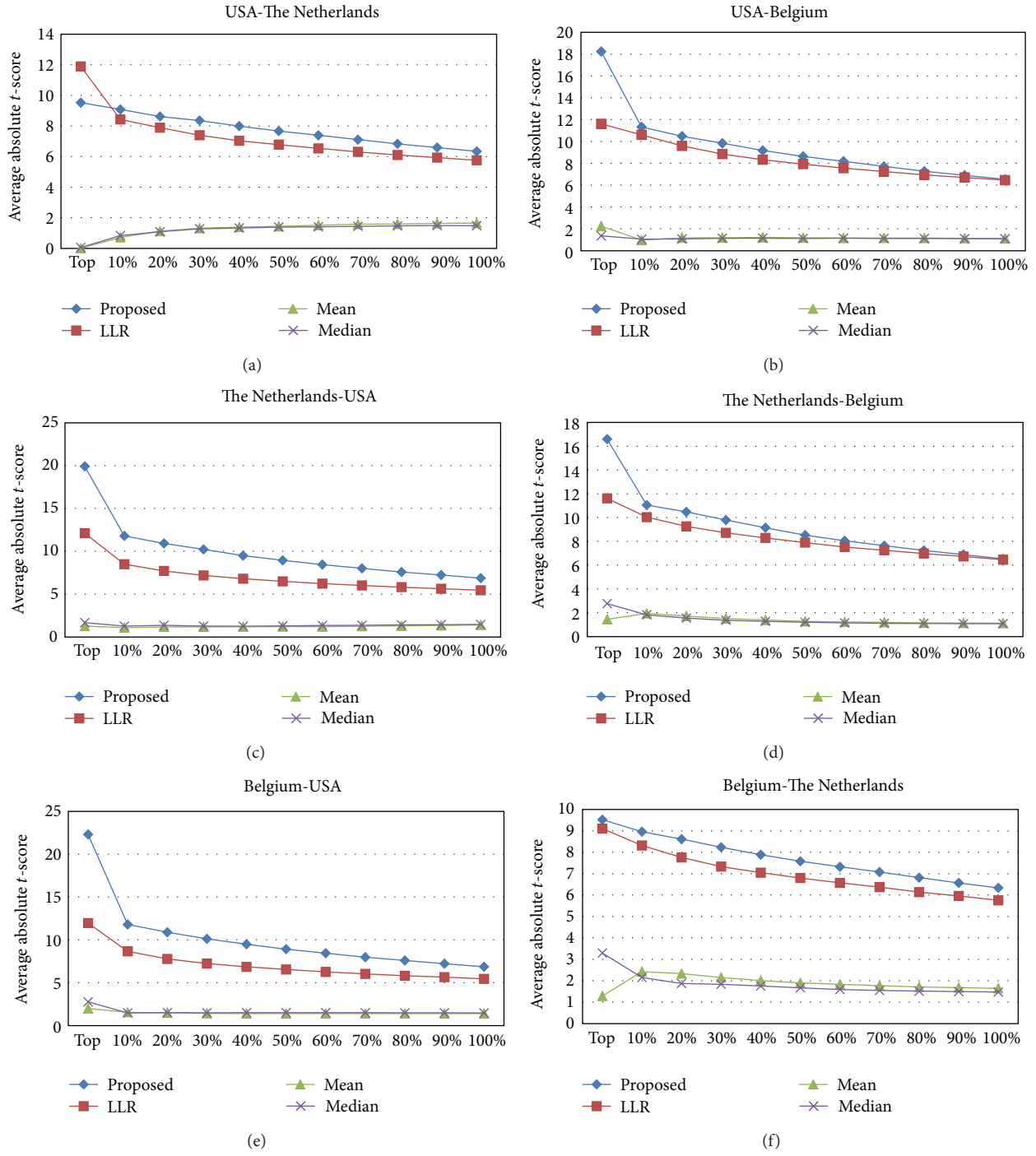


FIGURE 3: Discriminative power of pathway markers across different datasets. The pathway markers have been ranked and sorted using the first dataset, and their discriminative power has been reevaluated using the second dataset. As before, the mean absolute t-score was used for assessing the discriminative power.

marker evaluation and the remaining fold was used for feature selection. The second dataset was randomly divided into fivefolds, where fourfolds were used to train the LDA classifier, using the features selected from the first dataset, and the remaining fold was used to evaluate the classification performance. The overall setup is shown in Figure 4(b). To obtain reliable results, we repeated this experiment for 100 random partitions (of the second dataset) and report the average AUC as the performance metric. For these experiments, we used the three largest breast cancer datasets (USA, The Netherlands, and Belgium) among the six.

The results of the cross-dataset classification experiments are shown in Figure 6. As we can see from this figure, the proposed rank-based inference scheme typically outperforms

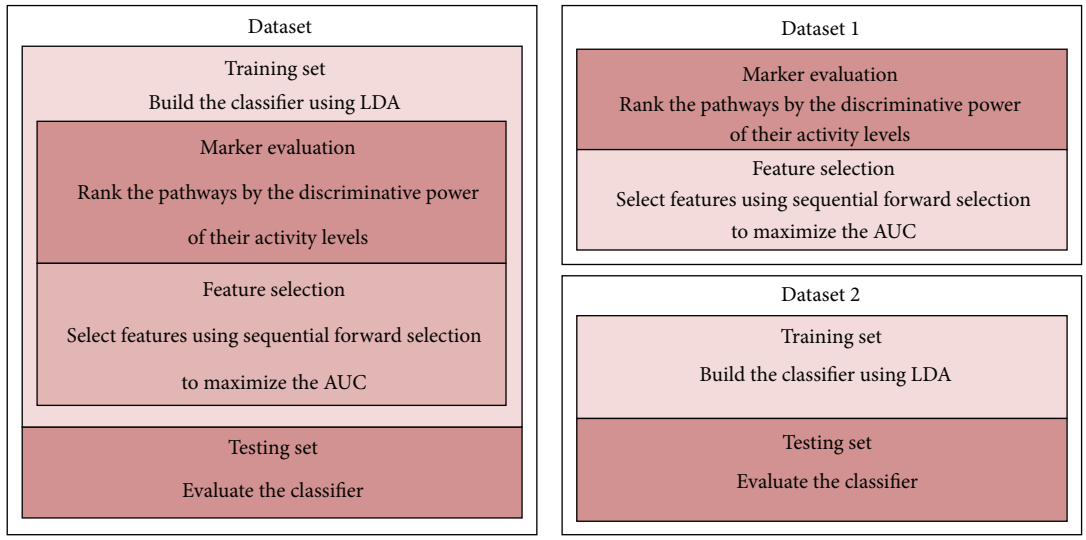


Figure 4: Experimental setup for evaluating the classification performance. (a) The setup for the within-dataset experiment. (b) The setup for the cross-dataset experiment.

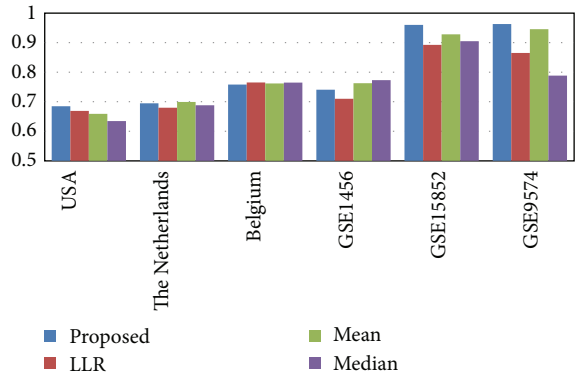


Figure 5: Classification performance for within-dataset experiments. The bars show the classification performance (average AUC) of different pathway activity inference methods evaluated on various breast cancer datasets.

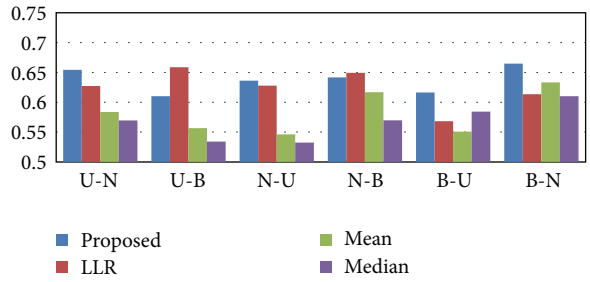


Figure 6: Classification performance for cross-dataset experiments. The bars show the cross-dataset classification performance (average AUC) of different pathway activity inference methods. The first dataset was used for selecting the pathway markers and the second dataset was used for training and evaluation of the classifier. The three largest breast cancer datasets were used: USA (U), The Netherlands (N), and Belgium (B).

other methods in terms of reproducibility. Furthermore, we can also observe that the proposed method yields consistent classification performance across experiments, while the performance of other inference methods is much more sensitive on the choice of the dataset. Next, we repeated the cross-dataset classification experiments based on the USA and the Belgium datasets after normalizing the raw data using RMA, GCRMA, and MAS5. As shown in Figure 7, the proposed method yields consistently good performance, regardless of the normalization method that was used.

Finally, we performed additional cross-dataset experiments after normalizing the USA and the Belgium datasets using different normalization methods. These results are summarized in Figures S6 and S7. We can see that the proposed pathway activity inference scheme is relatively robust to "normalization mismatch." Moreover, these results also show that the proposed scheme overcomes the problem of the previous LLR-based scheme [10] when used with GCRMA (see Figures 7, S6, and S7).

4. Conclusions

In this work, we proposed an improved pathway activity inference scheme, which can be used for finding more robust and reproducible pathway markers for predicting breast cancer metastasis. The proposed method integrates two effective strategies that have been recently proposed in the field: namely, the probabilistic pathway activity inference method [10] and the ranking-based relative gene expression analysis approach [3]. Experimental results based on several breast cancer gene expression datasets show that our proposed inference method identifies better pathway markers that have higher discriminative power, are more reproducible, and can lead to better classifiers that yield more consistent performance across independent datasets.

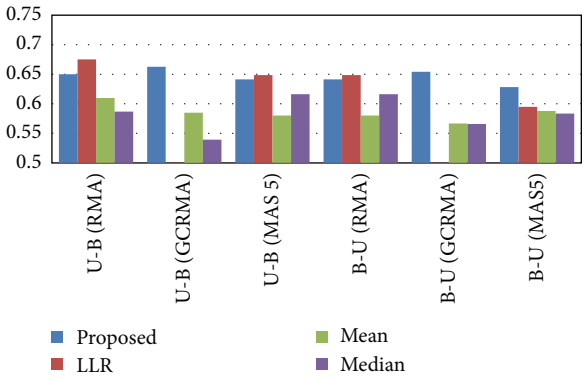


FIGURE 7: Classification performance for cross-dataset experiments. We repeated the cross-dataset experiments based on the USA and the Belgium datasets after normalizing the raw data using different normalization methods.

Acknowledgment

N. Khunlertgit has been supported by a scholarship from the Royal Thai Government.

References

[1] M. West, C. Blanchette, H. Dressman et al., "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 20, pp. 11462–11467, 2001.

[2] 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.

[3] D. Geman, C. D'Avignon, D. Q. Naiman, and R. L. Winslow, "Classifying gene expression profiles from pairwise mRNA comparisons," *Statistical Applications in Genetics and Molecular Biology*, vol. 3, no. 1, article 19, 2004.

[4] Y. Wang, J. G. M. Klijn, Y. Zhang et al., "Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer," *The Lancet*, vol. 365, no. 9460, pp. 671–679, 2005.

[5] L. Tian, S. A. Greenberg, S. W. Kong, J. Altschuler, I. S. Kohane, and P. J. Park, "Discovering statistically significant pathways in expression profiling studies," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 38, pp. 13544–13549, 2005.

[6] Z. Guo, T. Zhang, X. Li et al., "Towards precise classification of cancers based on robust gene functional expression profiles," *BMC Bioinformatics*, vol. 6, article 58, 2005.

[7] C. Auffray, "Protein subnetwork markers improve prediction of cancer outcome," *Molecular Systems Biology*, vol. 3, article 141, 2007.

[8] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker, "Network-based classification of breast cancer metastasis," *Molecular Systems Biology*, vol. 3, article 140, 2007.

[9] E. Lee, H. Y. Chuang, J. W. Kim, T. Ideker, and D. Lee, "Inferring pathway activity toward precise disease classification," *PLoS Computational Biology*, vol. 4, no. 11, Article ID e1000217, 2008.

[10] J. Su, B. J. Yoon, and E. R. Dougherty, "Accurate and reliable cancer classification based on probabilistic inference of pathway activity," *PloS ONE*, vol. 4, no. 12, Article ID e8161, 2009.

[11] J. Su, B. J. Yoon, and E. R. Dougherty, "Identification of diagnostic subnetwork markers for cancer in human protein-protein interaction network," *BMC Bioinformatics*, vol. 11, no. 6, article 8, 2010.

[12] J. A. Eddy, L. Hood, N. D. Price, and D. Geman, "Identifying tightly regulated and variably expressed networks by Differential Rank Conservation (DIRAC)," *PLoS Computational Biology*, vol. 6, no. 5, Article ID e1000792, 2010.

[13] N. Khunlertgit and B. J. Yoon, "Finding robust pathway markers for cancer classification," in *Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS '12)*, 2012.

[14] 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," *New England Journal of Medicine*, vol. 347, no. 25, pp. 1999–2009, 2002.

[15] C. Desmedt, F. Piette, S. Loi et al., "Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the TRANSBIG multicenter independent validation series," *Clinical Cancer Research*, vol. 13, no. 11, pp. 3207–3214, 2007.

[16] Y. Pawitan, J. Bjohle, L. Amler, and A. L. Borg, "Gene expression profiling spares early breast cancer patients from adjuvant therapy: derived and validated in two population-based cohorts," *Breast Cancer Research*, vol. 7, pp. R953–R964, 2005.

[17] H. Y. Chang, D. S. A. Nuyten, J. B. Sneddon et al., "Robustness, scalability, and integration of a wound-response gene expression signature in predicting breast cancer survival," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 10, pp. 3738–3743, 2005.

[18] R. Edgar, M. Domrachev, and A. E. Lash, "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository," *Nucleic Acids Research*, vol. 30, no. 1, pp. 207–210, 2002.

[19] R. C. Gentleman, V. J. Carey, D. M. Bates et al., "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, no. 10, p. R80, 2004.

[20] A. Liberzon, A. Subramanian, R. Pinchback, H. Thorvaldsdóttir, P. Tamayo, and J. P. Mesirov, "Molecular signatures database (MSigDB) 3.0," *Bioinformatics*, vol. 27, no. 12, pp. 1739–1740, 2011.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Interscience, New York, NY, USA, 2006.

[22] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2013, Article ID 953814, 12 pages
http://dx.doi.org/10.1155/2013/953814

Review Article

An Overview of the Statistical Methods Used for Inferring Gene Regulatory Networks and Protein-Protein Interaction Networks

Amina Noor,[1] Erchin Serpedin,[1] Mohamed Nounou,[2] Hazem Nounou,[3] Nady Mohamed,[4] and Lotfi Chouchane[4]

[1] *Electrical and Computer Engineering Department, Texas A&M University, College Station, TX 77843-3128, USA*
[2] *Chemical Engineering Department, Texas A&M University at Qatar, 253 Texas A&M Engineering Building, Education City, P.O. Box 23874, Doha, Qatar*
[3] *Electrical Engineering Department, Texas A&M University at Qatar, 253 Texas A&M Engineering Building, Education City, P.O. Box 23874, Doha, Qatar*
[4] *Department of Genetic Medicine, Weill Cornell Medical College in Qatar, P.O. Box 24144, Doha, Qatar*

Correspondence should be addressed to Amina Noor; amina@neo.tamu.edu

Received 20 July 2012; Revised 12 January 2013; Accepted 17 January 2013

Academic Editor: Yufei Huang

The large influx of data from high-throughput genomic and proteomic technologies has encouraged the researchers to seek approaches for understanding the structure of gene regulatory networks and proteomic networks. This work reviews some of the most important statistical methods used for modeling of gene regulatory networks (GRNs) and protein-protein interaction (PPI) networks. The paper focuses on the recent advances in the statistical graphical modeling techniques, state-space representation models, and information theoretic methods that were proposed for inferring the topology of GRNs. It appears that the problem of inferring the structure of PPI networks is quite different from that of GRNs. Clustering and probabilistic graphical modeling techniques are of prime importance in the statistical inference of PPI networks, and some of the recent approaches using these techniques are also reviewed in this paper. Performance evaluation criteria for the approaches used for modeling GRNs and PPI networks are also discussed.

1. Introduction

Postgenomic era is marked by the availability of a deluge of genomic data and has, thus, enabled the researchers to look towards new dimensions for understanding the complex biological processes governing the life of a living organism [1–5]. The various life sustaining functions are performed via a collaborative effort involving DNA, RNA, and proteins. Genes and proteins interact with themselves and each other and orchestrate the successful completion of a multitude of important tasks. Understanding how they work together to form a cellular network in a living organism is extremely important in the field of molecular biology. Two important problems in this considerably nascent field of computational biology are the inference of gene regulatory networks and the inference of protein-protein interaction networks. This paper first looks at how the genes and proteins interact with themselves and then discusses the inference of an integrative cellular network of genes and proteins combined.

Gene regulation is one of the many fascinating processes taking place in a living organism whereby the expression and repression of genes are controlled in a systematic manner. With the help of the enzyme RNA polymerase, DNA transcribes into mRNA which may or may not translate into proteins. It is found that in certain special cases mRNA is reverse-transcribed to DNA. The processes of transcription and translation are schematically represented in Figure 1, where the interactions in black show the most general framework and the interactions depicted in red occur less frequently. Transcription factors (TFs), which are a class of proteins, play the significant role of binding onto the DNA and thereby regulate their transcription. Since the genes may be coding for TFs and/or other proteins, a complex network of genes and proteins is formed. The level of activity of a gene

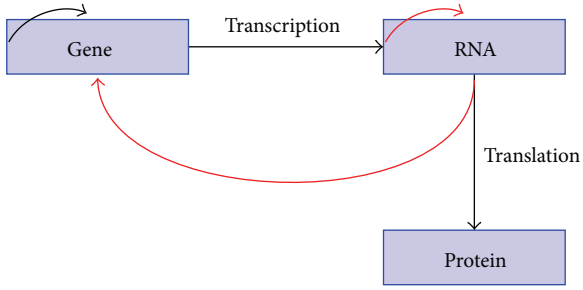


Figure 1: Central dogma of molecular biology.

is measured in terms of the amount of resulting functional product, and is referred to as gene expression. The recent high-throughput genomic technologies are able to measure the gene expression values and have provided large-scale data sets, which can be used to obtain insights into how the gene networks are organized and operated. One of the most encountered representations of gene regulatory networks is in terms of a graph, where the genes are depicted by its nodes and the edges represent the interactions between them.

The gene regulatory network (GRN) inference problem consists in understanding the underlying system model [6–10]. Simply stated, given the gene expression data, the activation or repression actions by a set of genes on the other genes need to be identified. There are several issues associated with this problem, including the choice of models that capture the gene interactions sufficiently well, followed by robust and reliable inference algorithms that can be used to derive decisive conclusions about the network. The inferred networks vary in their sophistication depending on the extent and accuracy of the prior knowledge available and the type of models used in the process. It is also important that the gene networks thus inferred should possess the highly desirable quality of reproducibility in order to have a high degree of confidence in them. A sufficiently accurate picture of gene interactions could pave the way for significant breakthroughs in finding cures for various genetic diseases including cancer.

Protein-protein interactions (PPIs) are of enormous significance for the workings of a cell. Insights into the molecular mechanism can be obtained by finding the protein interactions with a high degree of accuracy [11, 12]. The protein interaction networks not only consist of the binary interactions, rather, in order to carry out various tasks, proteins work together with cohorts to form protein complexes. It should be emphasized that a particular protein may be a part of different protein complexes, and hence the inference problem is much more complicated. The existing high-throughput proteomic data sets enable the inference of protein-protein interactions. However, it is found that the protein-protein interactions obtained by using different methods may not be equivalent, indicating that a large number of false positives and negatives are present in the data. Similar to the representation of gene regulatory networks, protein-protein interaction networks will also be modeled in terms of graphs, where the proteins denote the nodes and the edges signify whether an interaction is present between the adjacent nodes.

Many statistical methods have been applied extensively to solve various bioinformatics problems in the last decade. There are several papers that provide excellent review of various statistical and computational techniques for inferring genomic and proteomic networks [2, 12]. However, it is important to understand the fundamental similarities and differences that characterize the two inference problems. This paper provides an overview of the most recent statistical methods proposed for the inference of GRNs and PPI networks. For gene network inference, three large classes of modeling and inferencing techniques will be presented, namely, probabilistic graphical modeling approaches, information theoretic methods, and state-space representation models. Clustering and probabilistic graphical modeling methods which comprise the largest class of statistical methods using PPI data are reviewed for the protein-protein interaction networks. Through a concise review of these contemporary algorithms, our goal is to provide the reader with a sufficiently rich understanding of the current state-of-the-art techniques used in the field of genomic and proteomic network inference.

The rest of this paper is organized as follows. Section 2 describes some of the data sets available for the inference of genomic and proteomic networks. Section 3 reviews the recent statistical methods employed to infer gene regulatory networks. Protein-protein network inferencing techniques are reviewed in Section 4. The methods for obtaining an integrated network with gene network and protein-protein as subnetworks are given in Section 5. The inferred network evaluation is discussed in Section 6. Finally, conclusions are drawn in Section 7.

2. Available Biological Data

The postgenomic era is distinguished by the availability of huge amount of biological data sets which are quite heterogenous in nature and difficult to analyze [3]. It is expected that these data sets can aid in obtaining useful knowledge about the underlying interactions in gene-gene and protein-protein networks. This section reviews some of the main types of data used for the inference of genomic and proteomic networks, including, gene expression data, protein-protein interaction data, and ChIP-chip data.

2.1. Gene Expression Data. Of all the available datasets, gene expression data is the most widely used for gene regulatory network inference. Gene expression is the process that results in functional transcripts, for example, RNA or proteins, while utilizing the information coded on the genes. The level of gene expression is an important indicator of how active a gene is and is measured in the form of gene expression data. Similarity in the gene expression profiles of two genes advocates some level of correlation between them. In this paper, the gene expression data is denoted by means of a random variable $\mathbf{x}(t)$, where t stands for the time index.

2.1.1. cDNA-Microarray Data. One way of generating cDNA-microarray data is via the DNA microarray technology, which

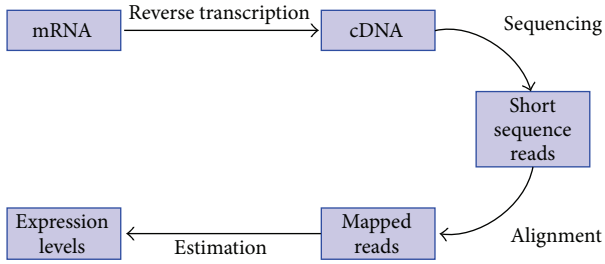


Figure 2: Expression estimation in RNA-Seq.

is by far the most popular method employed for this purpose. The number of data samples is in general much smaller than the number of genes. A main drawback associated with cDNA-microarray data is the noise in the observed gene expressions. Although the gene expression values should be continuous, the inability to measure them accurately suggests the use of discretized values.

2.1.2. RNA-Seq Data. The recent advancement of sequencing technologies has provided the ability to acquire more accurate gene expression levels [13]. RNA-Seq is a novel technology for mapping and quantifying transcriptomes, and it is expected to replace all the contemporary methods because of its superiority in terms of time, complexity, and accuracy. The gene expression estimation in RNA-Seq begins with the reverse transcription of RNA sample into cDNA samples, which undergo high-throughput sequencing, resulting in short sequence reads. These reads are then mapped to the reference genome using a variety of available alignment tools. The gene expression levels are estimated using the mapped reads, and several algorithms have been proposed in the recent literature to find efficient and more accurate estimates of the gene expression levels. This process is summarized in Figure 2. The gene expression data obtained in this manner has been found to be much more reproducible and less noisy as compared to the cDNA microarrays. The next subsection describes the data used for PPI network inference.

2.2. Protein-Protein Interaction Data. Large-scale PPI data have been produced in recent years by high-throughput technologies like yeast two-hybrid and tandem affinity purification, which provide stable and transient interactions, and mass spectrometry, which indicates the protein complexes [11, 12]. These data sets, in addition to being incomplete also consist of false positives, and, therefore, the interactions found in various data sets may not agree with each other. Owing to this disagreement, it is imperative to make use of statistical methods to infer the PPI networks by finding reliable and reproducible interactions and predict the interactions not found yet in the currently available data.

2.3. ChIP-Chip Data. ChIP-chip data, which is an abbreviation of chromatin immunoprecipitation and microarray (chip), investigates the interactions between DNA and proteins. This data provides information about the DNA-binding proteins. Since some of the genes encode for transcription factors (TFs) which in turn regulate some other genes and/or proteins, this information comes in hand for the inference of gene networks [10] and the integrated network. However, generating the ChIP-chip data for large genome would be technically and financially difficult.

2.4. Other Data Sets. Apart from the data sets described above, gene deletion and perturbation data are worth mentioning here. Perturbation data set is generated by performing an initial perturbation and then letting the system to react to it [14]. The gene expression values at the following time instants and at steady-state are measured, thereby obtaining the response of the genes to the specific perturbation which could be the increase or decrease of the expression level of all or certain genes. Gene deletion dataset, as the name indicates, involves deleting a gene and measuring the resulting expression level of other genes. This data may effectively uncover simple direct relationships [14].

3. Modeling and Inferring Gene Regulatory Networks

Gene regulatory networks capture the interactions present among the genes. Accurate and reliable estimation of gene networks is significantly crucial and can reap far-reaching benefits in the field of medicinal biology, for example, in terms of developing personalized medicines. The following subsections review the main statistical methods used for inference of gene regulatory networks. First, the important class of probabilistic graphical models is presented.

3.1. Probabilistic Graphical Modeling Techniques. Probabilistic graphical models have emerged as a useful tool for reverse engineering gene regulatory networks. A gene network is represented by a graph $\mathbf{G} = (V, E)$, where V represents the set of vertices (genes), and E denotes the set of edges connecting the vertices. The vertices of the graph are modeled as random variables and the edges signify the interaction between them. The expression value of gene i is denoted by X_i, and the total number of genes in the network is denoted by N. The following subsections briefly describe some of the robust and popular graphical modeling techniques for gene network inference.

3.1.1. Bayesian Networks. Bayesian networks model the gene regulatory networks as directed acyclic graphs (DAGs). To simplify the inference process, the probability distribution of DAG-networks is generally factored in terms of the conditional distributions of each random variable given its parents:

$$P(\mathbf{X}) = \prod_{i=1}^{N} P\left(X_i \mid Pa\left(X_i\right)\right), \tag{1}$$

where $Pa(X_i)$ denotes the parent of node X_i. The gene regulatory network is inferred by using the Bayesian network learning techniques. This is done by maximizing the

probability $P(\mathbf{G} \mid \mathbf{D})$, where $\mathbf{D}$ denotes the available gene expression data. Several scoring metrics have been proposed to obtain the best graph structure [15]. The network, thus, obtained is unique to the extent of equivalence class; that is, the independence relationships are uniquely identified.

The gene expression data available to date consist of very few data points, while the number of genes is substantially larger, rendering the system to be underdetermined. As an alternative to finding the complete networks, scientists have proposed looking at certain important features, for example, Markov relations and order relations. If a gene X is present in the minimal network blanketing the gene Y, then a Markov relation is said to be established. A relationship between two genes is referred to as an ordered relation if a particular gene X appears to be a parent of another gene Y in all the equivalent networks. By aggregating this information, it is possible to infer the underlying regulatory structure robustly and reliably. The network structure inferred in this manner looks at the static interactions only. In order to cater for the dynamic interactions inherent in gene networks, dynamic Bayesian networks (DBNs) have been used [16, 17].

3.1.2. Qualitative Probabilistic Networks. A novel method of modeling gene networks is via the usage of qualitative probabilistic networks (QPNs), which represent the qualitative analog of the DBNs [18]. The structural and independence properties of QPNs are the same as those of Bayesian networks. However, instead of being concerned about the local conditional probabilities of the random variables, the former class of models looks at how the changes in probabilities of the random variables affect the probabilities of their immediate parents. This change is measured in qualitative terms instead of quantitative values, that is, whether the probabilities increase, decrease, or stay the same as shown in Figure 3.

Two important properties of QPNs are the qualitative influences and the qualitative synergies. A positive influence denoted by $I^{+}(X, Y)$ indicates the greater possibility of Y having a higher value when that of X is high and vice versa, irrespective of all other variables; that is,

$$I^{+}(X, Y) \quad \text{iff } P(y \mid x, W) > P(y \mid -x). \tag{2}$$

In the case of three variables, QPNs look at the synergies. A positive additive synergy, denoted by $S^{+}(\{X, Y\}, Z)$, exists when the combined effect of the parent nodes is greater on the child node than their individual effects given by

$$S^{+}(\{X, Y\}, Z) \quad \text{iff } P(z \mid x, y, W) + P(z \mid -x, -y, W) > P(z \mid x, -y, W) + P(z \mid -x, y, W). \tag{3}$$

QPNs, thus, provide more insight into the gene networks by indicating whether a particular gene is a promoter or an inhibitor.

3.1.3. Graphical Gaussian Models. Graphical Gaussian models, also known as covariance selection or concentration graph models, provide a simple and effective way of characterizing the gene interactions [19, 20]. This method relies on assessing the conditional dependencies among genes in terms of partial correlation coefficients among the gene expressions and results in an undirected network. A covariance matrix is estimated using the available gene expression data sets. Suppose that $\mathbf{X} \in \mathbb{R}^{n \times n}$ denotes the gene expression data matrix, where the rows correspond to observations and the columns correspond to genes, then an estimate of the covariance matrix is obtained by

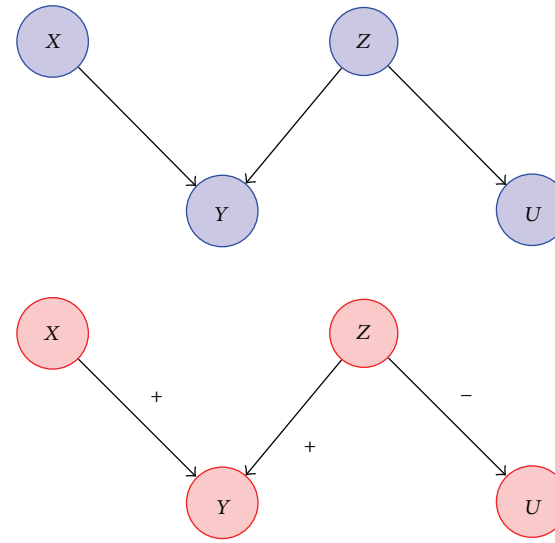


Figure 3: Qualitative probabilistic network (red) for a Bayesian network (blue).

$$\widehat{\mathbf{W}} = \frac{1}{N-1}\mathbf{X}^{T}\mathbf{X}. \tag{4}$$

Assuming invertibility of $\widehat{\mathbf{W}}$, the partial correlations can be determined as

$$\widehat{\rho}_{ij} = -\frac{\widehat{w}_{ij}}{\sqrt{\widehat{w}_{ii}\widehat{w}_{jj}}}, \tag{5}$$

where $\widehat{\rho}_{ij}$ denotes the partial correlation between genes i and j.

3.1.4. Graphical LASSO Algorithm. A major drawback of the covariance-matrix-estimation-based methods is their unreliability due to the small number of data samples. Making use of the fact that gene networks are inherently sparse, it is possible to obtain the dependencies between genes by means of a penalized linear regression approach [20]. The graphical Least Absolute Shrinkage and Selection Operator (LASSO) algorithm solves the network inference problem efficiently by maximizing the following penalized likelihood function:

$$\frac{2}{n}l(\mathbf{W}) = \log(\det(\mathbf{W})) - \operatorname{trace}\left(\widehat{\mathbf{W}}\mathbf{W}\right) - \rho\|\mathbf{W}\|_{1}, \tag{6}$$

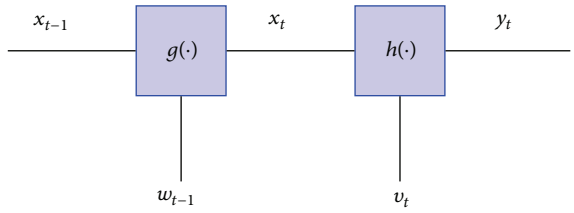


FIGURE 4: State-Space model.

where ρ controls the sparsity of the network, notation $||\cdot||_1$ represents the l_1-norm, and $\mathbf{W}$ denotes the covariance matrix. This minimization can be carried out by using block gradient descent methods, the details of which can be found in [20] and the references therein.

3.2. State-Space Representation Models. One of the earliest and widely used methods of modeling gene networks is by employing the state-space representation models [21]. As opposed to other classes, all the methods belonging to this class model the dynamic evolution of the gene network. These models generally consist of two sets of equations, the first set of equations representing the evolution of the hidden state variables denoted by $\mathbf{z}(t)$, and the second set of equations relating the hidden state variables with the observed gene expression data, denoted by $\mathbf{x}(t)$ as depicted in Figure 4. The functions $g(\cdot)$ and $h(\cdot)$ describe the evolution of hidden and observed variables, respectively. Next, in this section we will describe various models for gene network inference using the state-space representation model.

3.2.1. Linear State-Space Model. The simplest model for state-space equations is the linear Gaussian model given by [21, 22]:

$$\begin{aligned} \mathbf{z}(t) &= \mathbf{A}\mathbf{z}(t-1) + v(t), \\ \mathbf{x}(t) &= \mathbf{C}\mathbf{z}(t) + \mathrm{w}(t), \end{aligned} \tag{7}$$

where $\mathbf{A}$ is a matrix representing the regulatory relations between the genes, and t stands for the discrete time points. Difference equations are used in place of differential equations because discrete observations are available in the gene expression data. The noise components $v(t)$ and $\mathbf{w}(t)$ represent the system and the measurement noise, respectively, and are assumed to be Gaussian. The noise models the uncertainty present in the estimated gene expression data. The matrix $\mathbf{C}$ is generally considered to be an identity matrix. Inference in gene networks modeled by the state-space representation (7) can be performed using standard Kalman filter updates. The simplicity of the state-space model avoids overfitting of the network, and therefore, it provides reliable results.

3.2.2. Nonlinear Models. While it is useful to represent gene networks by simple models to ease the computational complexity, it is also imperative to incorporate nonlinear effects into the system equations, since the genes are known to interact nonlinearly [23]. A particular function that is frequently used to capture the nonlinear effects is the sigmoid squash function defined below in (9) [24]. The nonlinear state-space representation model capturing the gene interactions is described by the following system of equations:

$$\mathbf{z}(t) = \mathbf{A}\mathbf{z}(t-1) + \mathbf{B}f\left(\mathbf{z}(t-1), \boldsymbol{\mu}\right) + \mathbf{I}_0 + v(t), \tag{8}$$

where the jth entry of vector function $f(\cdot)$ is given by the sigmoid squash function:

$$f_j\left(z_j, \mu_j\right) = \frac{1}{1 + e^{-\mu_j z_j}}, \tag{9}$$

where μ is a parameter to be identified. Matrix $\mathbf{A}$ represents the linear relationships between the genes, while matrix $\mathbf{B}$ characterizes the nonlinear interactions. The problem, thus, boils down to the estimation of the following unknowns in the system:

$$\theta = \left[\mathbf{A}, \mathbf{B}, \boldsymbol{\mu}, \mathbf{I}_0\right], \tag{10}$$

where $\mathbf{I}_0$ models the constant bias. One way of solving these equations is by using the extended Kalman filter (EKF) [24], which is a popular algorithm for solving nonlinear state-space equations. EKF algorithm provides the solution by approximating the nonlinear system by its first-order linear approximation. Other variants of Kalman filter algorithm like the cubature Kalman filter (CKF), unscented Kalman filter (UKF), and particle filter algorithm are also used to solve such inference problems [25].

However, for many studies, the considered nonlinear model is comprised of a large number of unknowns and in order to estimate these unknown variables with considerable accuracy, data sets consisting of a large number of samples are required. The availability of smaller data sets represents an insurmountable obstacle in the reliable estimation of a large number of unknowns. This problem can be partially avoided by simplifying the model to include only nonlinear terms, and thus reducing the number of unknown parameters to the bare minimum [25] and by approximating μ to be one. The system of equations corresponding to such a parsimonious scenario is then given by

$$\mathbf{z}(t) = \mathbf{B}f\left(\mathbf{z}(t-1)\right) + v(t), \tag{11}$$

where f is the function defined previously.

3.2.3. Models with Sparsity Constraints. A crucial feature for many gene networks is their inherent sparsity; that is, all genes in the network are connected to a few other genes only. Therefore, matrices $\mathbf{A}$ and $\mathbf{B}$ depicting the regulatory relations between the genes are expected to contain only very few nonzero values as compared to the size of these matrices. Therefore, one may apply shrinkage-based methods like LASSO [25, 26] for parameter estimation and parsimonious model selection. One of the ways for inferring models with sparsity constraints is to perform dual estimation, which involves estimating the states and the parameters one by one. The hidden states can be estimated using the particle filter algorithm, and once all the estimates for the hidden states are obtained, they can be stacked together to form a matrix and

thus the following system of equations is obtained to perform the parameter estimation:

$$\begin{bmatrix} z_{n1} \\ z_{n2} \\ \vdots \\ z_{nI} \end{bmatrix} = \begin{bmatrix} f(z_{0,1}) & \cdots & f(z_{0,N}) \\ f(z_{1,1}) & \cdots & \vdots \\ \vdots & \ddots & \\ f(z_{I-1,1}) & & f(z_{I-1,N}) \end{bmatrix} \begin{bmatrix} b_{n1} \\ b_{n2} \\ \vdots \\ b_{nN} \end{bmatrix} + \begin{bmatrix} v_{n1} \\ v_{n2} \\ \vdots \\ v_{nI} \end{bmatrix}, \tag{12}$$

which can be expressed compactly in vector/matrix-form representation as

$$\mathbf{z}_n = \mathbf{\Phi}\mathbf{b}_n + v_n. \tag{13}$$

LASSO operates on this system of equations and produces a parameter vector $\mathbf{b}_n$ by minimizing the criterion [27]:

$$\min_{\mathbf{b}_n} \frac{1}{2}\left\|\mathbf{z}_n - \mathbf{\Phi}\mathbf{b}_n\right\|_2^2 + \rho\left\|\mathbf{b}_n\right\|_1. \tag{14}$$

The parameter estimates obtained using LASSO-based algorithms appear to be more reliable than the estimates provided by other approaches [25].

3.2.4. State-Space Models for Time-Delayed Dependencies. The state-space models discussed so far do not consider time delays whereas it has been found that time-delayed interactions are present in gene networks [28] due to the time required for the processes of transcription and translation to take place. One of the ways to model this phenomenon is by adopting the following state-space model:

$$\begin{aligned} \mathbf{z}(t) &= \mathbf{A}\mathbf{z}(t-1) + \mathbf{B}\mathbf{u}(t-\tau) + v(t), \\ \mathbf{x}(t) &= \mathbf{C}\mathbf{z}(t) + \mathbf{w}(t). \end{aligned} \tag{15}$$

In this state-space model, the input is considered to be the expression profile of a regulator such as a transcription factor. Here, $\mathbf{A}$ stands for the $N \times N$ state transition matrix, while $N \times p$ matrix $\mathbf{B}$ captures the effect of p regulators on the system. The value of the time delay τ is obtained by finding the best fit over a range of possible values using Akaike's information criterion (AIC) in order to avoid overfitting the network.

3.3. Information Theoretic Methods. Information theoretic methods have provided some of the most robust and reliable algorithms for gene network inference and form the basis of a standard in this field [29–31]. A particular advantage associated with these methods is their ability to work with minimal assumptions about the underlying network. This is in contrast with the probabilistic graphical modeling techniques as well as the state-space models, both of which have their own set of assumptions. As highlighted previously, a Markov network provides an undirected network, while Bayesian networks are not able to incorporate cycles or feedback loops. State-space models apart from the linear Gaussian model make critical assumptions on the model structure. These drawbacks are not present in the case of information theoretic methods. The following discussion presents the main information theoretic approaches for inferring gene regulatory networks.

3.3.1. Finding the Correlation between Genes. Two of the most fundamental concepts in information theory are mutual information and entropy. Mutual information between two random variables X and Y is defined as [32]

$$\begin{aligned} I(X;Y) &= \sum_{x,y}\left[p(x,y)\log\frac{p(x,y)}{p(x)\,p(y)}\right] \\ &= H(X) + H(Y) - H(X,Y), \end{aligned} \tag{16}$$

where H denotes the entropy or the uncertainty present in a random variable, and it is given by

$$H(X) = -\sum_{x} p(x)\log p(x). \tag{17}$$

Mutual information measures the correlation between two random variables. In the context of gene network inference, a higher mutual information between two genes indicates a higher dependency, and therefore, a possible interaction between them. Some of the most important and robust algorithms for gene network inference make use of the mutual information for finding the interacting genes [29, 30].

3.3.2. Identifying Indirect Interactions between Genes. If the mutual information between two genes is greater than a certain threshold, it indicates some correlation between them. However, this information alone is not sufficient to decide whether the genes are connected directly or indirectly via an intermediate gene. The data processing inequality (DPI) provides some insight to assess whether such a scenario holds. In case of three genes forming a Markov chain as shown in Figure 5, DPI can be expressed as

$$I(X;Y) \leq \min\left[I(X;Z), I(Y;Z)\right]. \tag{18}$$

Using this inequality, it is found that the interaction with the least mutual information is an indirect one. This method is employed in ARACNE [29], which has become a standard algorithm for gene network inference. However, DPI fails to hold in situations where one of the three genes is a parent gene to the other two genes. Conditional mutual information has been proposed to be used in such cases [30]. Conditional mutual information is defined as

$$\begin{aligned} I(X;Y \mid Z) &= \sum_{X,Y,Z}\left[p(x,y,z)\log\frac{p(x,y \mid z)}{p(x \mid z)\cdot p(y \mid z)}\right] \\ &= H(X,Z) + H(Y,Z) - H(Z) - H(X,Y,Z). \end{aligned} \tag{19}$$

If $I(X;Y \mid Z)$ is much less than $I(X;Y)$, it implies that Z is a parent of the genes X and Y as shown in Figure 5. In case the two quantities are almost equal, it means that the gene Z does not have any influence on the other two genes. Therefore, by employing the idea of conditional mutual information, indirect interactions in the case of common cause can be sifted.

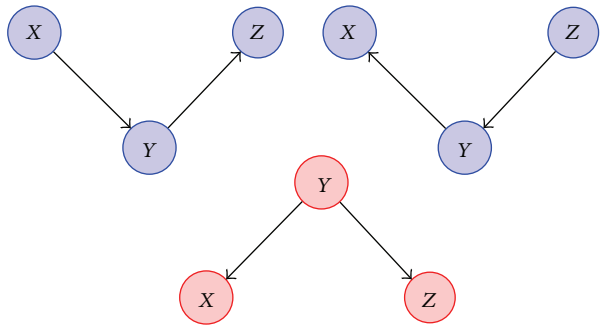


Figure 5: Markov chain (blue) and common cause (red).

3.3.3. Finding the Directed Networks. Calculating the mutual information using static data does not provide any information about the directed relationships. On the other hand, using time series data may indicate the directionality of interactions as well [33]. Mutual information for time series data can be expressed as

$$I\left(X_{t+1}; Y_t\right) = \sum_{x_{t+1}, y_t} \left[p\left(x_{t+1}, y_t\right) \log \frac{p\left(x_{t+1}, y_t\right)}{p\left(x_{t+1}\right) p\left(y_t\right)} \right]. \tag{20}$$

If a high value is obtained for $I(X_{t+1}; Y_t)$, it signifies a directed relationship from gene Y to X. While using these methods, the determination of the significance threshold is of considerable importance and can be estimated based on the prior knowledge about the network.

The information theoretic quantities discussed so far are symmetric (or bidirectional) and do not provide any information about the directionality by themselves. Some new metrics have been proposed recently to infer asymmetric or one-directional relationships such as the ϕ-mixing coefficient defined as [34]:

$$\phi\left(Y \mid X\right) = \max_{S \subseteq A, T \subseteq B} \left| \Pr\left\{Y \in T \mid X \in S\right\} - \Pr\left\{Y \in T\right\} \right|. \tag{21}$$

In other words, this coefficient provides a measure of independence or difference between two genes X and Y. DPI also holds true for the ϕ-mixing metric, and therefore, it can be used to identify the indirect interactions as in the case of mutual information.

3.3.4. Time-Delayed Dependencies. Another way of finding directed relationships is by detecting the time-delayed dependencies by using time series data. The time instants at which the mutual information goes above or drops below the thresholds τ_{up} and τ_{down}, respectively, are noted [35]. These instants are called the initial change of expression (IcE) times and are defined as

$$\text{IcE}\left(x_a\right) = \arg\min_j \left\{ \frac{x_a^j}{x_a^0} \geq \tau_{\text{up}} \text{ or } \frac{x_a^j}{x_a^0} \leq \tau_{\text{down}} \right\}. \tag{22}$$

It can be seen that a gene x_a can be a regulator for gene x_b if and only if (iff) $\text{IcE}(x_a) < \text{IcE}(x_b)$. The mutual information in this case is given by

$$I^k\left(x_a; x_b\right) = \sum_{i=1} \left[p\left(x_a^i, x_b^{i+k}\right) \log \frac{p\left(x_a^i, x_b^{i+k}\right)}{p\left(x_a^i\right) p\left(x_b^{i+k}\right)} \right], \tag{23}$$

where the delay is denoted by k. The next step consists in finding the maximum of the mutual information values calculated for all the time delays; that is,

$$I\left(x_a, x_b\right) = \max_k \left\{ I^k\left(x_a, x_b^{(k)}\right) \right\} \quad \text{for } k = 1, 2, \ldots, \text{while } \text{IcE}\left(x_a\right) \leq \text{IcE}\left(x_b\right). \tag{24}$$

If the value of the maximum mutual information is greater than a prespecified threshold, it is concluded that a directed relationship exists from x_a to x_b. The calculation of threshold is very important in all the information theoretic methods which is selected on the basis of the predetermined P-value [29]. This helps to obtain networks with the required significance value.

3.3.5. Model Selection. An important and necessary step in the implementation of the above-mentioned algorithms is the model selection. A network formed by using mutual information alone will result in an overfitted structure, and therefore, model selection becomes imperative. Minimum description length (MDL) principle was proposed as a general approach for model selection. MDL states that the network with the shortest coding length should be selected. For a network with a large number of nodes, the coding length will be large and vice versa. MDL principle provides a trade-off and aids in selecting only the significant interactions between the genes. MDL was applied in various ways in finding the coding length of the network and the probability densities associated with it [33]. Another way of using this principle is in conjunction with the maximum likelihood (ML) principle which results in a more general algorithm [36]. Further details on this algorithm can be found in [36]. Thus, it appears that the tools of information theory are quite powerful in modeling and inferring gene regulatory networks.

4. Inferring the Protein-Protein Interaction Networks

Having examined the gene network inference problem, this section describes the statistical methods that are used to find reliable and complete protein-protein interaction networks. As opposed to gene networks which are mostly inferred using the expression data or the likes of it, inference of PPI networks can be carried out in various ways such as phylogenetic profiling and identification of structural patterns. This paper focuses only on the methods that employ PPI data to make inference. The given data in this scenario are the protein-protein interactions. However, such data sets consist of a large number of false positives and negatives and are far from being complete and homogeneous. Therefore, only a small

overlap is found between the PPI data sets obtained from various sources. However, it is observed that the interactions predicted by more than one method are more reliable [37]. One of the challenges is the large number of interactions indicated by the PPI data as opposed to the considerably fewer interactions assumed to be present in reality. Therefore, the problem in this scenario is to find more reliable interactions and predict the yet unknown interactions. In addition, the protein interactions can be of different types ranging from stable ones to transient ones [37].

It is to be noted that as opposed to the gene networks, a lot of work can still be done for protein-protein network inference using the probabilistic methods. In a living organism, several proteins work together to carry out various tasks forming a protein complex. Most of the PPI data consists of binary interactions only and it is very rare to find interactions between more than two proteins simultaneously. Hence, identification of protein complexes is of prime importance to gain a better understanding of the cellular network.

Detecting protein complexes is a fundamental area of study of protein networks [38], for which various clustering methods were applied. One of the various ways of identifying the protein complexes include graph segmentation, where the graph is clustered into subgraphs using cost-based search algorithms. Another approach is broadly categorized as conservation across species [38], where alignment tools are used to find the complexes that are common in multiple data sets coming from different species. In what follows, some of the recently proposed probabilistic graphical-modeling- and clustering-based methods are described.

4.1. Markov Networks. The available PPI data look mostly at the binary interactions, and interactions of three or more genes are hard to find. However, it is important to look at the interacting proteins holistically. Markov networks are probabilistic graphical modeling techniques which result in undirected graphs. Suppose $\mathbf{X} = \{X_1, \ldots, X_N\}$ is a vector of random variables modeling the proteins. Their joint distribution is captured in terms of the potentials $\psi_c \in \Psi$. The random variables $\mathbf{X}_c$ that are connected to each other are called the scope for the particular potential ψ_c. The joint probability distribution is then given by

$$P(\mathbf{X} = \mathrm{x}) = \frac{1}{Z}\prod_{c \in C} e^{\psi_c(\mathbf{x}_c)}, \tag{25}$$

where Z is the normalizing constant also called the partition function. In this way, a compact representation of the probability distribution is obtained. The network structure is learned by using the independence properties of Markov networks using the available PPI data. The details of this method can be found in [37].

4.2. Bayesian Networks. Another way of modeling PPI networks is by means of Bayesian networks (BNs) [39], which represent a probabilistic graphical modeling technique. The inference algorithm is based on finding the conditional probability densities $P(X_i \mid C)$, where C denotes the class variable, and X_i denotes the ith node in the network. A particular strength of BNs is their ability to estimate model parameters even in the presence of incomplete data, which is often the case with the PPI networks. This fact makes BN a perfectly suited method for modeling protein networks. One way of estimating the model parameters is via the Expectation Maximization (EM) algorithm [39]. The joint probability distribution is expressed as

$$P(C, X_1, \ldots, X_N) = P(C)\prod_i P(X_i \mid C). \tag{26}$$

Assuming all the random variables to be independent of each other, the posterior density is given by

$$P(C \mid X_1, \ldots, X_N) = P(C)\prod_i \frac{P(X_i \mid C)}{P(X_i)}. \tag{27}$$

Once the model parameters are known, prediction can be made about random variables for which the data may not be available. Therefore, this algorithm provides a suitable method for finding protein complexes.

4.3. Graphical Clustering Methods. One of the ways of graph clustering is based on supervised learning [12, 38]. The subgraphs are modeled using Bayesian networks, and the features consist of topological patterns of graphs and biological properties. Rather than assuming the widely used cliqueness property, which considers all the nodes to be connected with each other, the algorithm looks for the properties that are inferred from already known complexes. Two important features are the label C indicating whether a subgraph is a complex and the number of nodes N. The other feature descriptors including degree statistics, graph density, and degree correlation statistics are indicated by $X_1, \ldots, X_m$ and are considered independent given C and N. The number of nodes in and off itself is an important feature. Its importance can be seen from the fact that a larger number of nodes in a subgraph indicate a lesser probability of it being a clique. All the subgraphs are assigned scores by making use of these properties. One way of finding how probable it is for a subgraph to be a protein complex is to perform simple hypothesis testing by calculating the following conditional probability [12, 38]:

$$\begin{aligned} L &= \log \frac{p(c_1 \mid x_1, \ldots, x_m)}{p(c_0 \mid x_1, \ldots, x_m)} \\ &= \log \frac{p(n \mid c_1)\prod_{k=1}^{m} p(x_k \mid n, c_1)}{p(n \mid c_0)\prod_{k=1}^{m} p(x_k \mid n, c_0)}, \end{aligned} \tag{28}$$

where the posterior probabilities are calculated via Bayes rule as

$$\begin{aligned} &p(c_i \mid n, x_1, \ldots, x_m) \\ &\quad = \frac{p(n, x_1, \ldots, x_m \mid c_i = 1)\, p(c_i = 1)}{p(n, x_1, \ldots, x_m)} \\ &\quad = \frac{p(x_1, \ldots, x_m \mid n, c_i = 1)\, p(n \mid c_i = 1)\, p(c_i = 1)}{p(n, x_1, \ldots, x_m)}. \end{aligned} \tag{29}$$

These probability densities can be calculated using maximum likelihood methods. By comparing the obtained score to a predetermined threshold, some of the subgraphs can be labeled to be complexes. This algorithm takes the weighted matrix of PPI data as input, where the weights are assigned using the likelihood of any particular interaction. Several other graphical-clustering-based methods are surveyed in [12].

4.4. Matrix Factorization Methods for Clustering. Nonnegative matrix factorization (NMF) is a method widely used in problems of clustering. Application of this technique has been proposed recently in [40], where an ensemble of nonnegative factored matrices obtained using protein-protein interaction data are combined together to perform soft clustering. The importance of this step lies in the fact that a particular object may belong to multiple classes. Hence, the various algorithms reported in the literature performing hard clustering may not be of much benefit in such scenarios. This ensemble NMF method is observed to classify the proteins in accordance with the functions they perform and also identify the multiple groups they belong to.

The algorithm produces τ base clusterings by factorizing the symmetric data matrix S of protein interactions in the following manner [40]:

$$\min_{\mathbf{V}>0} \left\| \mathbf{S} - \mathbf{V}\mathbf{V}^{\mathbf{T}} \right\|_F^2, \tag{30}$$

where $||\cdot||_F$ denotes the Frobenius norm. The factors $\mathbf{V}$ produced in this manner are not unique. Let k_i be the number of clusters in the ith base cluster, each with a different value in order to promote diversity. Once the ensemble of factored matrices is available, the next step is to construct the graph by combining the information present in them. Parameter $l = k_1 + \cdots + k_\tau$ gives the total number of basis vectors which are denoted by $\mathbf{V} = \{v_1, \ldots, v_l\}$. Each vector denotes a node on the graph, and the edge weight is calculated using the Pearson correlation for a pair of vector (v_i, v_j) given by

$$\operatorname{cor}\left(v_i, v_j\right) = \frac{1}{2}\left(\frac{\left(v_i - \overline{v}_i\right)^T \left(v_j - \overline{v}_j\right)}{\left\|v_i - \overline{v}_i\right\|_2 \cdot \left\|v_j - \overline{v}_j\right\|_2} + 1 \right). \tag{31}$$

Having looked at the GRNs and PPI network inference problems individually, we now proceed to review the recent advancements in the joint modeling of the two networks.

5. An Integrated Cellular Network

The advances in reverse engineering of GRNs and PPI networks have paved the way for joint estimation of GRNs and PPI networks [41]. This is a step towards the inference of an integrated network consisting of genes, proteins, and transcription factors, indicating interactions among themselves and each other. Figure 6 shows the schematic of an integrated cellular network. In this section, we review two important ways of estimating a joint network.

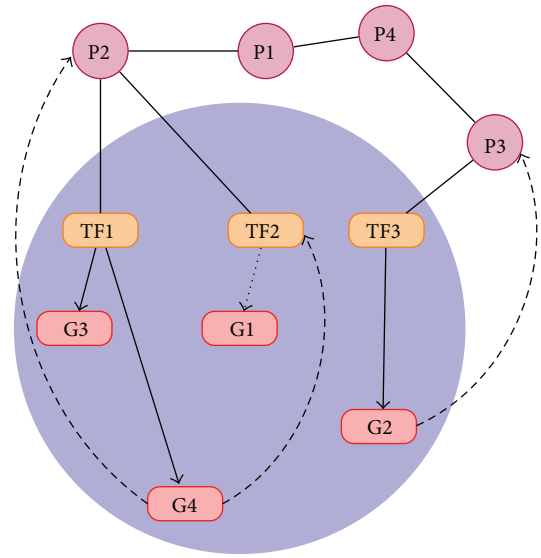


Figure 6: An integrated cellular network.

5.1. Probabilistic Graphical Models for Joint Inference. Reference [41] proposed an interesting method for estimating GRNs and PPI networks simultaneously. Suppose that the gene expression is denoted by $\mathbf{x}$ and PPI data is represented by $\mathbf{y}$. The algorithm provides an undirected protein network G_p and a directed gene network G_r, modeled using Markov and Bayesian networks, respectively, by maximizing their joint distribution; that is,

$$\begin{aligned} P\left(G_r, G_p \mid X, Y\right) &\propto P\left(G_r, G_p, X, Y\right) \\ &= P\left(X \mid G_r\right) P\left(Y \mid G_p\right) P\left(G_r, G_p\right), \end{aligned} \tag{32}$$

where $P(X \mid G_r, G_p) = P(X \mid G_r)$ and $P(Y \mid Gr, G_p) = P(Y \mid G_p)$. The inference on Markov and Bayesian networks is performed in the same manner as explained in the previous sections. The two subnetworks are estimated iteratively till the algorithm converges. Further details on this algorithm can be found in [41].

5.2. Joint Estimation Using State-Space Model. State-space model can also be used to obtain an integrated network of gene and protein-protein interactions [42, 43]. A novel approach employing nonlinear model is proposed in [43], where the system parameters are estimated using constrained leastsquares. The gene expression is assumed to follow a dynamic model given by

$$x_i\left(t+1\right) = x_i\left(t\right) + \sum_{j=1}^{N} a_{ij} s_i\left(t\right) - \lambda_i z_i\left(t\right) + k_i + w_i\left(t\right), \tag{33}$$

where

$$s_j\left(t\right) = f_i\left(y_j\left(t\right)\right) = \frac{1}{1 + \exp\left\{-\left(y_j\left(t\right) - \mu_j\right)/\sigma_j\right\}}, \tag{34}$$

and y_j denotes the protein activity profile of jth transcription factor, and its mean and standard deviations are represented by μ_j and σ_j, respectively. The magnitude of a_{ij} indicates the strength of relationship between the jth TF and ith gene, and the sign suggests whether it is an excitatory or inhibitory relationship. The model in (33) suggests that the gene expression level at tth time instant depends upon the gene expression level at the previous time instant as well as the protein activity level. The degradation effect of gene expression is modeled by λ_i, k_i is a constant representing the basal level, and $w_i(t)$ is the Gaussian noise modeling the uncertainties in the model and the errors in the data.

The protein activity level follows the following dynamic model:

$$y_n(t+1) = y_n(t) + \sum_{m=1}^{M} b_{nm} y_n(t)\, y_m(t) + \alpha_n x_n(t) - \beta_n y_n(t) + h_n + \nu_n(t), \tag{35}$$

where b_{ij} gives the relationship between the proteins, α_n indicates the translation effect of mRNA to protein, and $\nu_n(t)$ is the Gaussian noise. The unknown parameters for both the models are given by

$$\theta_i = \begin{bmatrix} a_{i1} & \cdots & a_{iN} & \lambda_i & k_i \end{bmatrix}^T, \qquad \phi_n = \begin{bmatrix} b_{n1} & \cdots & b_{nM} & \alpha_n & \beta_n & h_n \end{bmatrix}^T \tag{36}$$

and are estimated by solving a constrained least squares problem [43]. Once the individual subnetworks are obtained, they are merged together to form one cellular network with the TFs connecting them together.

The problem of inferring an integrated network is in relatively initial stages, and several avenues of research are still open. Moreover, comparison studies are needed so as to determine the merits and demerits of the different methods in use.

6. Performance Evaluation

The inference accuracy can be assessed using the knowledge of a gold-standard network or the true network. In order to benchmark the algorithms, the correctly identified edges or true positives (TPs) need to be calculated. In addition, the number of false positives (FPs), or the edges incorrectly indicated to be present, and false negatives (FNs) which is the missed detection should also be counted [10]. With these values in hand, true positive rate or recall; that is, TPR = TP/(TP+FN), false positive rate; that is, FPR = FP/(FP+TN), and positive predictive value; that is, PPV = TP/(TP + FP), also called the precision, can be calculated. These quantities enable us to view the performance graphically by the area under the ROC curve which plots FPR versus the TPR. These criteria are most widely used as the fidelity criterion for gene network inference algorithms.

While it is possible to identify the gene regulatory relationships experimentally, it would not only be technically prohibitive but also proved to be very costly. For this reason, several *in silico* and *in vivo* networks have been generated to assist in benchmarking the network inference algorithms. Foremost among these are the DREAM (dialogue on reverse engineering assessment and methods) [44] and IRMA (*in vivo* reverse engineering and modeling assessment) [45] datasets. Reference [10] provides a unified survey of some of the important algorithms in gene network inference algorithms using these datasets.

7. Discussions and Conclusions

This paper reviews the main statistical methods used for inference of gene and protein-protein networks. PPI network inference can be carried out in a wide variety of ways by exploiting phylogenetics information and sequencing data. This paper focused only on those inference methods that employ PPI data.

For the inference of gene regulatory networks, the problem can be simply stated as follows: given the gene expression data, find the interactions between the genes. Three major classes of statistical methods were reviewed in this paper: probabilistic graphical models, state-space models, and information theoretic methods. For all these methods, modeling as well as inferencing techniques was discussed. It is observed that much progress has been made in the field of GRN inference. However, almost all of the proposed network inference methods in the literature work with only the popular gene expression data sets. An interesting part of future work could be integrating different data sets and biological knowledge available to come up with better and more robust algorithms.

Comparing the three broad classes of statistical methods reviewed in the paper, it is found that the information theoretic methods have advantages over the other methods in terms of minimal modeling assumptions and, therefore, are capable of modeling more general networks. Graphical modeling techniques assume the network to be acyclic in case of Bayesian network modeling and provide an undirected graph when using Markov networks. The state-space nonlinear models work with nonlinear functions which may not be the true representative of the underlying network, thereby resulting in less robust algorithms.

In case of PPI network prediction, the most popular statistical method is clustering. In addition, probabilistic graphical modeling techniques are also used. However, several important avenues of research are still open. Since the Markov networks and Bayesian networks are able to model PPI networks efficiently, other probabilistic graphical techniques such as factor graphs could potentially be used for solving this inference problem. Clustering methods are more suited to the PPI network inference problem as the main emphasis is on the identification of protein complexes. It is found that certain important and popular modeling techniques may fail to model PPI networks [46]. Also, clustering methods based on mutual information could be used [47].

Several statistical methods have been proposed to infer an integrated network of transcription regulation and protein-protein interaction. A state-space model for integrated network inference involves parameter estimation which

indicates the strength of the inhibitory and excitatory regulations. As the cellular networks are known to be sparse, employing sparsity-constrained least squares for parameter estimation as proposed in [25] is expected to result in more robust inference algorithms.

Recent years have shown tremendous and rapid progress in the field of cellular network modeling. With the amount and types of data sets increasing, algorithms combining multiple datasets are necessary for future.

Acknowledgments

This paper was made possible by QNRF-NPRP Grant no. 09-874-3-235 and support from NSF Grant no. 0915444. The statements made herein are solely the responsibility of the authors.

References

[1] X. Zhou and S. T. C. Wong, *Computational Systems Bioinformatics*, World Scientific, 2008.

[2] Y. Huang, I. M. Tienda-Luna, and Y. Wang, "Reverse engineering gene regulatory networks: a survey of statistical models," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 76–97, 2009.

[3] X. Zhou, X. Wang, and E. R. Dougherty, *Genomic Networks: Statistical Inference from Microarray Data*, John Wiley & Sons, 2006.

[4] H. Kitano, "Computational systems biology," *Nature*, vol. 420, no. 6912, pp. 206–210, 2002.

[5] B. Mallick, D. Gold, and V. Baladandayuthapani, *Bayesian Analysis of Gene Expression Data*, Wiley, 2009.

[6] H. D. Jong, "Modeling and simulation of genetic regulatoy systems: a literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, 2002.

[7] X. Cai and X. Wang, "Stochastic modeling and simulation of gene networks," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 27–36, 2007.

[8] H. Hache, H. Lehrach, and R. Herwig, "Reverse engineering of gene regulatory networks: a comparative study," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2009, Article ID 617281, 2009.

[9] F. Markowetz and R. Spang, "Inferring cellular networks—a review," *BMC Bioinformatics*, vol. 8, article S5, 2007.

[10] C. A. Penfold and D. L. Wild, "How to infer gene networks from expression profiles, revisited," *Interface Focus*, vol. 3, pp. 857–870, 2011.

[11] J. Wang, M. Li, Y. Deng, and Y. Pan, "Recent advances in clustering methods for protein interaction networks," *BMC Genomics*, vol. 11, no. supplement 3, article S10, 2010.

[12] X. Li, M. Wu, C. K. Kwoh, and S. K. Ng, "Computational approaches for detecting protein complexes from protein interaction networks: a survey," *BMC Genomics*, vol. 11, no. 1, article S3, 2010.

[13] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-Seq," *Nature Methods*, vol. 5, no. 7, pp. 621–628, 2008.

[14] K. Y. Yip, R. P. Alexander, K. K. Yan, and M. Gerstein, "Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data," *PLoS ONE*, vol. 5, no. 1, Article ID e8121, 2010.

[15] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009.

[16] K. Murphy and S. Mian, "Modeling gene expression data using dynamic Bayesian networks," Tech. Rep., University of California, Berkeley, Calif, USA, 2001.

[17] Y. Zhang, Z. Deng, H. Jiang, and P. Jia, "Inferring gene regulatory networks from multiple data sources via a dynamic Bayesian network with structural EM," in *DILS*, S. C. Boulakia and V. Tannen, Eds., vol. 4544 of *Lecture Notes in Computer Science*, pp. 204–214, Springer, 2007.

[18] Z. M. Ibrahim, A. Ngom, and A. Y. Tawfik, "Using qualitative probability in reverse-engineering gene regulatory networks," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 2, pp. 326–334, 2011.

[19] N. Kramer, J. Schafer, and A. Boulesteix, "Regularized estimation of large-scale gene association networks using graphical gaussian models," *BMC Bioinformatics*, vol. 10, no. 1, p. 384, 2009.

[20] P. Menéndez, Y. A. I. Kourmpetis, C. J. F. ter Braak, and F. A. van Eeuwijk, "Gene regulatory networks from multifactorial perturbations using graphical lasso: application to the DREAM4 challenge," *PLoS ONE*, vol. 5, no. 12, Article ID e14147, 2010.

[21] F.-X. Wu, W.-J. Zhang, and A. J. Kusalik, "Modeling gene expression from microarray expression data with state-space equations," in *Pacific Symposium on Biocomputing*, R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung, and T. E. Klein, Eds., pp. 581–592, World Scientific, 2004.

[22] Z. Wang, F. Yang, D. W. C. Ho, S. Swift, A. Tucker, and X. Liu, "Stochastic dynamic modeling of short gene expression time-series data," *IEEE Transactions on Nanobioscience*, vol. 7, no. 1, pp. 44–55, 2008.

[23] M. Quach, N. Brunel, and F. D'alché-Buc, "Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference," *Bioinformatics*, vol. 23, no. 23, pp. 3209–3216, 2007.

[24] Z. Wang, X. Liu, Y. Liu, J. Liang, and V. Vinciotti, "An extended kalman filtering approach to modeling nonlinear dynamic gene regulatory networks via short gene expression time series," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 3, pp. 410–419, 2009.

[25] A. Noor, E. Serpedin, M. N. Nounou, and H. N. Nounou, "Inferring gene regulatory networks via nonlinear state-space models and exploiting sparsity," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1203–1211, 2012.

[26] A. Noor, E. Serpedin, M. Nounou, and H. Nounou, "Inferring gene regulatory networks with nonlinear models via exploiting sparsity," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '12)*, pp. 725–728, March 2012.

[27] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B*, vol. 58, pp. 267–288, 1996.

[28] C. Koh, F. X. Wu, G. Selvaraj, and A. J. Kusalik, "Using a state-space model and location analysis to infer time-delayed Regulatory Networks," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2009, Article ID 484601, 3 pages, 2009.

[29] A. A. Margolin, I. Nemenman, K. Basso et al., "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context," *BMC Bioinformatics*, vol. 7, no. supplement 1, article S7, 2006.

[30] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring connectivity of genetic regulatory networks using information-theoretic criteria," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, no. 2, pp. 262–274, 2008.

[31] A. Noor, E. Serpedin, M. N. Nounou, H. N. Nounou, N. Mohamed, and L. Chouchane, "Information theoretic methods for modeling of gene regulatory networks," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB '12)*, pp. 418–423, 2012.

[32] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley Interscience, 2006.

[33] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring gene regulatory networks from time series data using the minimum description length principle," *Bioinformatics*, vol. 22, no. 17, pp. 2129–2135, 2006.

[34] M. Vidyasagar, "Probabilistic methods in cancer biology," *Childhood*, vol. 20, pp. 82–89, 2011.

[35] P. Zoppoli, S. Morganella, and M. Ceccarelli, "TimeDelay-ARACNE: reverse engineering of gene networks from time-course data by an information theoretic approach," *BMC Bioinformatics*, vol. 11, no. 1, article 154, 2010.

[36] J. Dougherty, I. Tabus, and J. Astola, "Inference of gene regulatory networks based on a universal minimum description length," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2008, Article ID 482090, 2008.

[37] A. Jaimovich, G. Elidan, H. Margalit, and N. Friedman, "Towards an integrated protein-protein interaction network: a relational Markov network approach," *Journal of Computational Biology*, vol. 13, no. 2, pp. 145–164, 2006.

[38] Y. Qi, F. Balem, C. Faloutsos, J. Klein-Seetharaman, and Z. Bar-Joseph, "Protein complex identification by supervised graph local clustering," *Bioinformatics*, vol. 24, no. 13, pp. i250–i268, 2008.

[39] J. R. Bradford, C. J. Needham, A. J. Bulpitt, and D. R. Westhead, "Insights into protein-protein interfaces using a Bayesian network prediction method," *Journal of Molecular Biology*, vol. 362, no. 2, pp. 365–386, 2006.

[40] D. Greene, G. Cagney, N. Krogan, and P. Cunningham, "Ensemble non-negative matrix factorization methods for clustering protein-protein interactions," *Bioinformatics*, vol. 24, no. 15, pp. 1722–1728, 2008.

[41] N. Nariai, Y. Tamada, S. Imoto, and S. Miyano, "Estimating gene regulatory networks and protein-protein interactions of Saccharomyces cerevisiae from multiple genome-wide data," *Bioinformatics*, vol. 21, no. supplement 2, pp. ii206–ii212, 2005.

[42] C. W. Li and B. S. Chen, "Identifying functional mechanisms of gene and protein regulatory networks in response to a broader range of environmental stresses," *Comparative and Functional Genomics*, vol. 2010, Article ID 408705, 2010.

[43] Y. C. Wang and B. S. Chen, "Integrated cellular network of transcription regulations and protein-protein interactions," *BMC Systems Biology*, vol. 4, no. 1, article 20, 2010.

[44] http://wiki.c2b2.columbia.edu/dream.

[45] I. Cantone, L. Marucci, F. Iorio et al., "A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches," *Cell*, vol. 137, no. 1, pp. 172–181, 2009.

[46] R. Schweiger, M. Linial, and N. Linial, "Generative probabilistic models for protein-protein interaction networks-the biclique perspective," *Bioinformatics*, vol. 27, no. 13, pp. i142–i148, 2011.

[47] 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.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2013, Article ID 527295, 10 pages
http://dx.doi.org/10.1155/2013/527295

Research Article

MRMPath and MRMutation, Facilitating Discovery of Mass Transitions for Proteotypic Peptides in Biological Pathways Using a Bioinformatics Approach

Chiquito Crasto,[1] Chandrahas Narne,[2] Mikako Kawai,[3] Landon Wilson,[4] and Stephen Barnes[1,3,4,5]

[1] *Department of Genetics, University of Alabama at Birmingham, Birmingham, AL 35294, USA*
[2] *Department of Computer and Information Sciences, University of Alabama at Birmingham, Birmingham, AL 35294, USA*
[3] *Department of Pharmacology and Toxicology, University of Alabama at Birmingham, Birmingham, AL 35294, USA*
[4] *Centers for Nutrient-Gene Interactions, University of Alabama at Birmingham, Birmingham, AL 35294, USA*
[5] *Targeted Metabolomics and Proteomics Laboratory, University of Alabama at Birmingham, Birmingham, AL 35294, USA*

Correspondence should be addressed to Chiquito Crasto; chiquito@uab.edu

Received 23 September 2012; Revised 20 December 2012; Accepted 20 December 2012

Academic Editor: Erchin Serpedin

Quantitative proteomics applications in mass spectrometry depend on the knowledge of the mass-to-charge ratio (*m/z*) values of proteotypic peptides for the proteins under study and their product ions. MRMPath and MRMutation, web-based bioinformatics software that are platform independent, facilitate the recovery of this information by biologists. MRMPath utilizes publicly available information related to biological pathways in the Kyoto Encyclopedia of Genes and Genomes (KEGG) database. All the proteins involved in pathways of interest are recovered and processed *in silico* to extract information relevant to quantitative mass spectrometry analysis. Peptides may also be subjected to automated BLAST analysis to determine whether they are proteotypic. MRMutation catalogs and makes available, following processing, known (mutant) variants of proteins from the current UniProtKB database. All these results, available via the web from well-maintained, public databases, are written to an Excel spreadsheet, which the user can download and save. MRMPath and MRMutation can be freely accessed. As a system that seeks to allow two or more resources to interoperate, MRMPath represents an advance in bioinformatics tool development. As a practical matter, the MRMPath automated approach represents significant time savings to researchers.

1. Introduction

A feature of the last two decades of biomedical research has been the generation of "–omics" data, a result of the pursuit of *discovery*. The introduction of soft ionization techniques for analysis of peptides and proteins by mass spectrometry in the 1980s [1, 2] led to a plethora of applications related to the identification of proteins from a wide variety of proteomes, from microorganisms to plants to mammals. These studies largely defined the *measurable* peptidome and by implication the proteome. They were also designed to "discover" significant protein changes, such as abundance and modifications. Because of the complications resulting from multiple hypotheses testing, however, detecting differences between treated and control samples has often met with limited success [3]. Concern has been expressed, for example, over the failure of different participating laboratories to systematically determine the same proteins that distinguish cancer patients from controls [4].

The next phase of proteomics is moving towards targeted, hypothesis-driven experiments. It integrates knowledge from previous proteomics discovery endeavors (2D-gel/peptide mass fingerprinting, MuDPIT, and GeLC-tandem mass spectrometry), microarray analysis (DNA, mRNA and microRNA chips, as well as DNA deep sequencing and RNA Seq), from the general scientific literature (particularly signal transduction pathways), or from the detailed study of one or several proteins in a known complex or a biological pathway. Figure 1

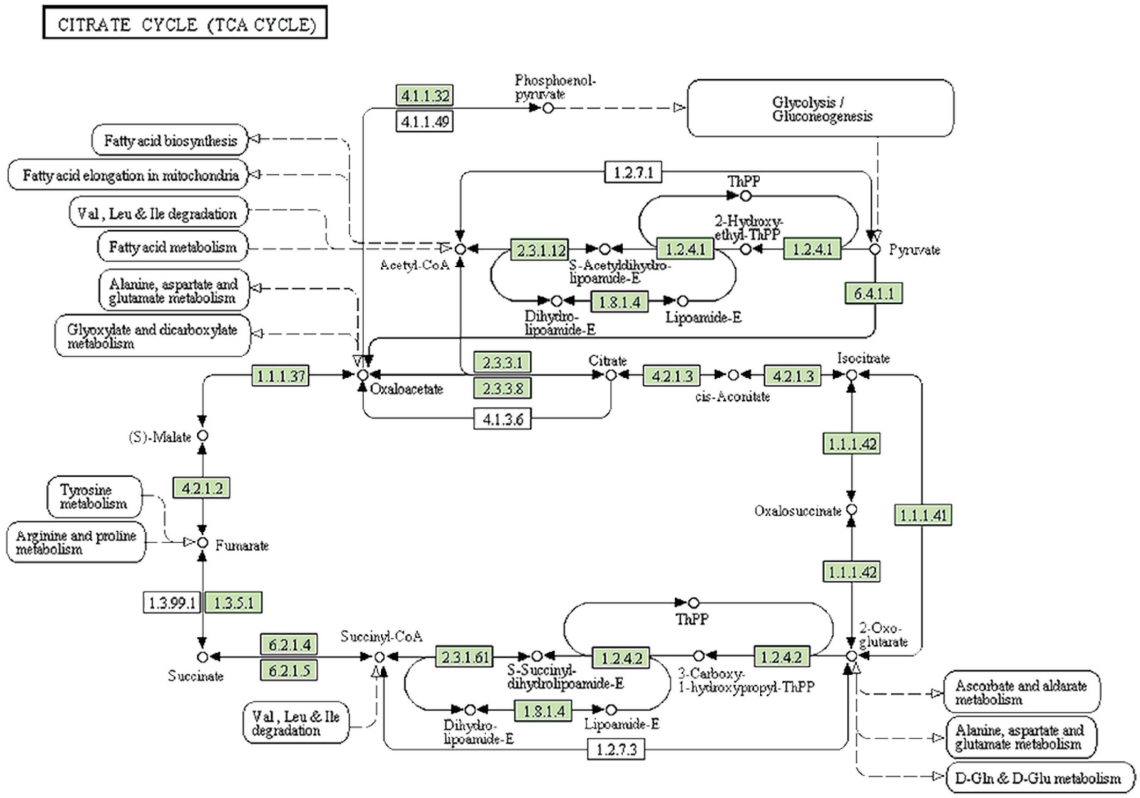


Figure 1: The figure represents the pathway for the citrate tricarboxylic acid cycle for humans as seen at the KEGG resource. The components, proteins, and reagents, highlighted in green, are those involved in the pathway for humans. The components not highlighted are part of the generic TCA cycle pathway. If another species is selected then only those components that contribute to the pathway are highlighted. The figure is a screen capture from the URL, http://www.genome.jp/kegg-bin/show_pathway?org_name=mmu&mapno=00062.

represents the tricarboxylic acid (TCA) cycle for humans as visualized through the KEGG (Kyoto Encyclopedia of Genes and Genomes: http://www.KEGG.jp/).

The development of targeted analysis of proteins has been facilitated by the use of multiple reaction ion monitoring (MRM), a mass spectrometry technique commonly used for the quantitative analysis of drugs and their metabolites [5]. If trypsin is used as the protease to cleave proteins into peptides, the resulting tryptic peptides, many of which consist of seven to 25 amino acids, would be suitable for analysis by MRM-MS on a triple quadrupole mass spectrometer. The molecular ion of the tryptic peptide ion (usually doubly charged) is selected by the first quadrupole, fragmented by collision with neutral gas in the second quadrupole, and specific, peptide sequence-dependent product ions are selected by the third quadrupole. The ion intensity resulting in this double selection process is typically measured for 20–30 msec at a time. Then another molecular ion/product ion combination representing a second tryptic peptide is examined. This process can be repeated 30–50 times a second before cycling back to the molecular ion/product ion combination for the tryptic peptide for the first protein. If the signal intensities of the analytes being measured are strong enough, the period for each channel can be shortened and as many as 500 transitions a second can be monitored.

The process of ensuring that a peptide from a specific protein in a biological pathway is proteotypic, especially when done manually, is prohibitive. It involves (1) selecting protein(s) involved in a biological (disease, metabolic, etc.) pathway, clicking on the source-link for the protein(s), and accessing the web page that contains information about the protein(s), or from other sources; (2) obtaining their amino acid sequences in the FASTA format; (3) submitting the sequences for *in silico* enzymatic digestion; (4) organizing the peptides that may be suitable for analysis, (5) carrying out a BLAST search for each.

Other attempts to identify suitable tryptic peptides for quantitative LC-MRM-MS analysis have either been based on a pragmatic approach (by inspection of peptide MS/MS data collected on instruments in investigators' laboratories), or on predictive tools based on a peptide training set. The latter increases the chance of selecting high intensity peptide ions. Skyline (https://brendanx-uw1.gs.washington.edu/labkey/project/home/software/Skyline/begin.view), a downloadable tool that performs many of the above steps, is available for investigators and end-users on the Windows platform. Skyline, for a given protein or peptide, calculates the masses of tryptic peptides and their fragment ions and includes provisions to filter out oxidizable groups and to allow biochemical posttranslational modifications. While Skyline is an excellent tool for investigators with experience in the use and applications of peptide mass spectrometry, it nonetheless represents a barrier for biologists who have identified areas of biochemistry where identification of changes (by western

blotting or DNA/RNA measurements) in a critical element of a pathway prompts a thorough investigation of all the components of the pathway and in some cases neighboring pathways. MRMPath and MRMutation allow the biologist to accurately recover the peptide and associated mass spectral data about the proteins in the biological pathway(s) so that the information can be transferred to the domain of mass spectrometry.

The present study therefore has two goals: (1) to create a freely-accessible, web-based software tool (MRMPath), that is, Internet browser accessible and hence not subject to dependence on the computer operating system. This software would dynamically retrieve and process known pathways of metabolism and protein signaling using an automated bioinformatics approach. Peptides that can be evaluated for their proteotypic character using BLAST searches would be extracted using this software. Any investigator would be able to access MRMPath and download and store results; and (2) the creation of a second web-based tool (MRMutation) that dynamically accesses all the known mutations (germline, somatic, and experimental) of a given protein in order to identify those tryptic peptides which would contain the mutation.

2. Methods

The proteins associated with human diseases and other processes, including metabolic and cellular processes in many species in which the genome sequences are known, are cataloged in a publicly available web resource, the Kyoto Encyclopedia of Genes and Genomes (KEGG) (URL http://www.genome.jp/kegg/). This resource is well visited by researchers the world over. It provides information related to pathways which are categorized according to metabolic, genetic information processing, environmental information processing, cellular processes, organismal systems, and human diseases. The individual pathways include the complement of proteins that are involved in them. Figure 1 is a screen capture, for the TCA (tricarboxylic acid) cycle in humans, presented through KEGG's web interface. The result of a user query in the KEGG for a specific pathway is a generic, non clickable representation of all the components involved in a pathway. For a pathway, a user can then choose the one appropriate for each species catalogued by KEGG. When a user chooses a species, for example, *Homo sapiens*, the proteins and other components that are specifically involved in the pathway for humans become "live" or clickable links. In Figure 1, these boxes are colored green.

For proteins and enzymes (which are represented using the Enzyme Commission nomenclature), the user is taken to a page where additional information is available related to the protein, which includes alternative nomenclature for that protein in other resources, the DNA sequence from which the protein sequence is intuitively translated, the family from which the protein arises, and the link to that protein in the FASTA format (which is accessed in MRMPath), among other information.

MRMPath facilitates the collection of the protein amino acid sequence data presented by KEGG. It is freely available on the Internet (http://tmpl.uab.edu/MRMPath/). Only an Internet browser is needed to access and use MRMPath. The system was designed for free use, mitigating the need for platform-specific computer operating systems, or to download and install software.

On accessing MRMPath and clicking on the "MRMPath" link, a user is offered three choices that involve deploying MRMPath by (1) processing proteins involved in a pathway stored in KEGG; (2) processing a protein from EXPASY (formerly, SWISSPROT) by entering the EXPASY protein ID; (3) cutting and pasting the protein sequence into a text box. Figure 2, a screen capture of the MRMPath home page, illustrates the choices that are available for protein sources.

2.1. MRMPath and KEGG. The first option allows users to use a drop-down menu to select among the pathways that are available in KEGG. When a user clicks on the pathway of choice, the system automatically populates a second drop-down menu which identifies only the species for which the selected pathway is available in KEGG. When the user clicks the "Submit" button, the system automatically downloads and represents the pathway to the investigator just as the user would see in KEGG, that is, with only the components (proteins) from the pathway highlighted in green as they are relevant to the species. This pathway is downloaded dynamically from the KEGG web resources and presented to the MRMPath user as a virtual webpage (precluding the need to store information); its HTML (hypertext markup language) webpage is processed and modified (on the UAB servers). The page illustrating the biological pathway for a species appears to the user exactly as it would appear to a KEGG user. The links for specific "live" components—proteins involved in the pathway for that species—are changed such that clicking on these links now deploys the MRMPath software for that protein, instead of leading to the webpage that contains additional information for that protein in KEGG.

2.2. MRMPath Processing. The amino acid sequence for the protein involved in the pathway is recovered in the FASTA format and subjected to *in silico* trypsin digestion (MRMPath also allows users to perform digestions using chymotrypsin, Arg-C, Lys-C, and Glu-C). Following a tryptic digest, cleavage occurs on the C-terminal side of arginine and lysine residues except when the next amino acid is proline. The mass-to-charge (m/z) values for the monoisotopic, doubly charged tryptic peptides are determined from the empirical formula for each peptide residue, using the elemental masses for carbon (12.00000000), hydrogen (1.00782503), nitrogen (14.00307401), and oxygen (15.99491462) [6]. Peptides with less than seven amino acids or more than 25 amino acids are not considered. Peptides containing cysteine or methionine resides are filtered out because of modifications that may arise from nonbiological events during sample processing. These peptides thus filtered are processed to calculate m/z values for b-ions and y-ions. Typically, these are larger than the m/z of the doubly charged molecular ion. In general, peptides chosen for MS/MS are doubly or triply charged; therefore, their higher mass, singly charged product ions

Figure 2: Front page of the Targeted Metabolomics and Proteomics Laboratory website. This is the home page for MRMPath and MRMutation. The three input choices for MRMPath—processing of peptides of proteins involved in biological pathways via KEGG, through Accession IDs in UniProt and direct input of a protein sequence—are illustrated.

have *m/z* values that could not arise from a singly charged molecular ion at the same *m/z* value as the doubly or triply charged peptide ion. The filtered tryptic amino acid sequences are subjected to automated BLAST analysis at NCBI (http://blast.ncbi.nlm.nih.gov/Blast.cgi). This is carried out either on a single tryptic peptide or on all the selected tryptic peptides for a given protein.

The following specific example illustrates the specific data-mining steps that the system deploys following a user query. When a user chooses a pathway and a species, MRMPath automatically creates a URL (Universal Resource Locator) which a user would otherwise manually type to access that pathway for that species. For example, consider the link http://www.genome.jp/kegg-bin/show_pathway?org_name=mmu&mapno=00062. The organism's name is identified by the three-letter species code "mmu" (*Mus musculus*). The numerical representation "0062" refers to the pathway, "fatty acid elongation." In KEGG, this pathway is represented under the category "Metabolism," and subcategory "Lipid Metabolism." We initially recovered and stored the codes for all the organisms and pathways in KEGG.

One of the components of this pathway is the mitochondrial trans-2-enoyl-CoA reductase (EC: 1.3.1.38). Within KEGG, when this component is clicked, it takes the user to information about that enzyme in KEGG through the URL (http://www.genome.jp/dbget-bin/www_bget?mmu:26922). Through MRMPath, when the pathway is downloaded and processed, each link within the downloaded file is modified. The KEGG link for mitochondrial trans-2-enoyl-CoA reductase would be automatically modified

to http://www.genome.jp/dbget-bin/www_bget?-f+-n+a+mmu:26922, which is the link to the FASTA file for this protein. The FASTA formatted protein sequence is then further processed.

Leveraging pathways published at the KEGG resource is an innovative aspect of MRMPath. MRMPath can process proteins involved in pathways and makes them all available to mass spectrometry specialists.

For the two other deployment strategies available in MRMPath, the above description is the same, except that only one protein at a time is processed.

2.3. EXPASY. The EXPASY Bioinformatics Resource Portal (http://www.expasy.org/) contains the world's most comprehensive and highly curated repository for proteins. In addition to information related to protein sequences, this resource is constantly updated with tools and subdata bases for different aspects of the analysis of proteins. The database within EXPASY that stores and allows access to proteins is UniProtKB (http://www.uniprot.org/). This resource allows access to proteins on the database through a keyword search or a descriptor search or by using the UniProt accession ID.

MRMPath uses web-accessibility techniques that were discussed previously to download a pathway or a FASTA-formatted protein sequence from KEGG. For example, consider a protein with a UniProt Accession ID, P47888. Clicking on the link associated with this Accession ID, http://www.uniprot.org/uniprot/P47888, allows a user to access additional information about this protein. MRMPath manipulates this link such that its algorithms can automatically access and download the FASTA formatted sequence from this protein through the webpage with the link, http://www.uniprot.org/uniprot/P47888. The sequence is thus downloaded and can be processed further by MRMPath, as described in the section on MRMPath and KEGG.

2.4. User Supplied Sequences. The third option, shown at the bottom of Figure 2, is a text box, which allows the user to cut and paste a protein sequence. This could be an entire protein as well as a fragment. This protein sequence is then processed through MRMPath in the same way as described when a FASTA formatted protein sequence is automatically downloaded from KEGG or from UniProt.

2.5. MRMPath's Process. As illustrated in the above section, the input for MRMPath is a protein or peptide sequence. This sequence can be automatically extracted for a protein involved in a biological pathway from KEGG, or a protein that is stored in the proteomics repository at EXPASY, or a user-supplied sequence. MRMPath processes a sequence as follows.

Enzymatic Digest. The sequence is first processed to create peptides following a theoretical enzymatic digest. MRMPath allows a user to choose between trypsin (cleaves on the carboxyl side of Arg and Lys), chymotrypsin (cleaves on the carboxyl side of Phe, Tyr, and Trp), AspN (cleaves the amino side of aspartate residues), GluC (cleaves the carboxyl side of aspartate and glutamate residues), LysC (cleaves on the carboxyl side of Lys residues), and ArgC (cleaves on the carboxyl side of Arg).

Selectivity for Met and Cys. The peptides obtained by the above user-determined enzymatic digest are processed to delete any that contain methionine or cysteine amino acid residues since these are susceptible to oxidation during sample processing and may not reflect the biology that is under investigation. A peptide containing methionine or cysteine may exist in several oxidized forms in addition to the unmodified peptide, rendering uncertainty in quantitative analysis.

Selection by Peptide Sizes. The resulting peptides are then filtered for the total number of amino acid residues. Only those peptides having seven or more, but not more than 25, residues are selected for further processing. Peptides with fewer than seven residues are unlikely to be proteotypic, whereas those with more than 25 residues become harder to detect or exceed the mass range of the quadrupole mass filter. The latter would have doubly charged ions that would be greater than m/z 1300.

Theoretical MS/MS Spectrum. For each of the resulting peptides from a protein, their multiply charged precursor ion can be collisionally dissociated producing b- and y-product ions. The y-ion masses (following cleavage at the amide bonds and containing the C-terminal residue) and b-ion masses (caused by cleaving the amide bonds and retaining from the N-terminal residue) are determined and tabulated. The m/z ratio for the peptide precursor ion is also calculated. These results are displayed on a browser following MRMPath's use (Figure 3).

2.6. BLAST Searching. Figure 4 shows that a BLAST search is also included for each fragment in the results of the webpage. When a user clicks this button, an automated BLAST search is initiated. The steps involved in the BLAST search are identical to a manual BLAST search for proteins on the webpages of NCBI BLAST (http://blast.ncbi.nlm.nih.gov/). During the manual search, a user will choose the type of BLAST, protein, nucleotide, and so forth, against a specific genomics resource (RefSeq—http://www.ncbi.nlm.nih.gov/RefSeq/ e.g.). A prompt appears on the webpage indicating after how long the search is likely to be completed. The results are then presented pictorially with color codes indicating the closeness of the BLAST results. The color red indicates a high similarity, the color black indicates lower than 40 percent similarity.

In MRMPath's automated BLAST search, the process is similar to the manual web-based BLAST search (http://blast.ncbi.nlm.nih.gov/), however, without manually entering the protein sequence (using NCBI's unique identifying number for the protein, the FASTA- or free-formatted protein sequence). MRMPath's BLAST search occurs in two steps. First, following a BLAST request for a peptide, the algorithm creates a URL. Embedded into this URL are query parameters: these include the peptide sequence, number of hits to return, and data source to search (NCBI). MRMPath then scans the NCBI BLAST server to identify a Process

Click here to download this into an Excel sheet
NOTE: Please click on the 'YES' button if a warning appears when you try to open the excel sheet

hsa:3417 IDH1, IDCD, IDH, IDP, IDPC, PICD; isocitrate dehydrogenase 1 (NADP+), soluble (EC:1.1.1.42); K00031 isocitrate dehydrogenase [EC:1.1.1.42] (A)

BLAST ALL FRAGMENTS

Sequence	m/z Parent Ion	B Ion Mass	Y > Parent Ions
BLAST IIWELIK	457.792	227.1760	801.4921
		413.2553	688.4080
		542.2979	502.3287
		655.3819	
		768.4660	
		896.5609	
BLAST LIFPYVELDLHSYDLGIENR	1203.6235	227.1760	2293.1551
		374.2444	2180.0710
		471.2971	2033.0026
		634.3605	1935.9499
		733.4289	1772.8865
		862.4715	1673.8181
		975.5556	1544.7755
		1090.5825	1431.6915
		1203.6666	1316.6645
		1340.7256	
		1427.7576	
		1590.8209	
		1705.8479	
		1818.9320	
		1875.9535	
		1989.0376	
		2118.0801	
		2232.1231	
		2388.2241	

Figure 3: A screen capture of MRMPath results (truncated) shows the peptides for a chosen protein (isocitrate dehydrogenase) from the TCA cycle pathway. The peptides are a result of a tryptic digest, the precursor ion values and the b- and y-ions whose masses are greater than that of the precursor ion identified. The link towards the top of the page allows users to download the processing result to an Excel spreadsheet. The buttons that allow BLAST searching of individual peptides as well all peptides from the chosen protein are also illustrated in the figure.

hsa:3417 IDH1, IDCD, IDH, IDP, IDPC, PICD; isocitrate dehydrogenase 1 (NADP+), soluble (EC:1.1.1.42); K00031 isocitrate dehydrogenase [EC:1.1.1.42] (A)

Sequence	Blast results
IIWELIK	This fragment is found only in protein **hsa:3417 IDH1, IDCD, IDH, IDP, IDPC, PICD; isocitrate dehydrogenase 1 (NADP+), soluble (EC:1.1.1.42); K00031 isocitrate dehydrogenase [EC:1.1.1.42] (A)** and does not show significant similarity with other proteins!
LIFPYVELDLHSYDLGIENR	This fragment is found only in protein **hsa:3417 IDH1, IDCD, IDH, IDP, IDPC, PICD; isocitrate dehydrogenase 1 (NADP+), soluble (EC:1.1.1.42); K00031 isocitrate dehydrogenase [EC:1.1.1.42] (A)** and does not show significant similarity with other proteins!

Figure 4: The result of a BLAST search of the tryptic peptide, the first peptide from Figure 3, shows that no results are found. The *m/z* for the parent ion of this peptide is also illustrated. If sequences similar to the peptide were identified, the top ten results would appear with links back to the GenBank resource for each similar sequence in the third column in the figure.

ID created by the BLAST system and the time it will take for the BLAST search to be completed. The program then automatically suspends processing for that amount of time (this might typically last from four to five seconds, longer if the BLAST servers are busy). After this "wait" time has elapsed, the program creates a second URL which includes the Process ID and dynamically extracts the BLAST results. One difference between the automated and manual searches is that results with very small sequence similarity will not be returned in the automated system as viable results.

The top ten BLAST results are returned to the user. If strong similarities are not found, then the program informs the user that the peptide does not have significant similarity with other proteins. The top ten BLAST results (if available) are then presented to the user in a webpage, with links to the sources of the proteins in GenBank.

2.7. Comprehensive Processing in MRMPath. In Figure 4, at the top right hand corner is a button, "BLAST ALL FRAGMENTS." This facility allows users to perform BLAST searches on all the peptides that result from the MRMPath filter at the same time. As has been explained previously, given that there is a wait time while BLAST searching occurs, users are likely to have to wait for several minutes for all BLAST searching on all the fragments to be complete. This process would be lengthened even more if done manually, where every fragment would have to be individually entered into the BLAST query "box."

When a user chooses to process a protein from a pathway through KEGG, MRMPath allows the option of processing all the proteins involved in a pathway for a particular species at the same time. Each protein involved in the pathway for a user-queried species will be processed in the same way as a single protein would, user-defined enzymatic digest and selection for peptide length (7–25 amino acid residues) and peptide sequences lacking methionine and cysteine residues.

In addition to the results being published on MRMPath's results webpage, where they can be downloaded, the results are also automatically written to an Excel spreadsheet. A new spreadsheet is generated with every MRMPath use. The same results that are available on the results web page are stored in the spreadsheet. We have placed, and continue to endeavor to place, information in the spreadsheet in the right format such that it can automatically, entered as input into the setup of manufacturer software such as Midas and MRMPilot for LC-MRM-MS analysis.

2.8. MRMutation. MRMutation was developed as a companion system to MRMPath, primarily because it involves several processing features that have been described for proteins (either cut and paste, or a protein from EXPASY, or proteins in a pathway stored in KEGG). MRMutation is available at the same resource that houses MRMPath. A user can deploy the software by entering a protein as its EXPASY Accession Number or by a descriptor for the protein, for example, TP53 human. In the latter case, the software accesses the EXPASY UniProt page for this entry.

The user must select the appropriate protein record. Protein accession numbers with a "P" as the first character are the most informative about the known mutations of the protein. The unique identifiers for each of these mutations are then retrieved and the information for the protein is processed. Processing involves subjecting each protein (obtained as the protein's FASTA formatted sequence) to a tryptic digest. This is to determine whether peptides with mutations are suited to multiple reaction ion monitoring. An output similar to the one for each protein in MRMPath is produced, except that it only contains the peptides that contain mutations. Figure 5 illustrates this table, where the mutated amino acid residue is highlighted. A table is generated that lists b- and y-ions whose masses are greater than the parent peptide ion m/z ratio. In addition, just as for results in MRMPath, the results of MRMutation are also available for download in a Microsoft Excel spreadsheet.

3. Discussion

The success of proteomics requires the development of informatics tools to enable the investigator to design targeted mass spectrometry experiments that answer specific biological hypotheses. Because of the immense amount of information involved, it has become important to create methods whereby biologists, in addition to mass spectrometry experts/operators, can contribute to the process. In the present study, MRMPath and MRMutation have successfully been put into practice to empower biologists to find those parts of a protein's sequence that are suitable for MRM analysis. These programs are also valuable to the mass spectrometry expert.

The value of the approach taken in creating MRMPath is driven by the extensiveness of the information available in the KEGG database. While manually searching for information on a single protein in this database is feasible, when confronted by 20–30 members of an entire pathway, it became obvious that an automated approach was necessary. MRMPath allows the investigator to select the pathway and the species of interest and then uses a data mining approach to filter information that is associated with each protein. Once captured, the protein sequence information is processed on local computers that automatically "cut" the protein into smaller peptide sequences obeying the biochemical rules set by the protease that would be used. Peptide analysis using the MRM approach is more specific for peptides that have seven or more amino acid residues. On the other hand much larger peptides (>25 amino acids) are harder to detect in most current mass spectrometers. The MRMPath software filters the peptides from a given protein to create a list of those that have 7–25 amino acids. There are many posttranslational modifications to proteins (and hence peptides) in biological and pathologic systems. Some of these, particularly oxidations, can occur after the sample has been taken and while it is being processed, in preparation for analysis. The sulfur atoms in cysteine and methionine residues are particularly prone to this—in the case of cysteine, many investigators block its free sulfhydryl group with an alkylating agent prior to analysis. Since controlling this oxidation is difficult and variable, MRMPath automatically filters out peptides that contain cysteine or methionine residues.

MRMPath software takes each filtered peptide and calculates the m/z of its precursor (molecular) ion and the product b- and y-ions. The latter are restricted to those that have values (singly charged) that are larger than the m/z of doubly charged precursor ion. The higher m/z values ensure that a singly charged precursor ion cannot contribute to the analysis of the doubly charged peptide.

It is important to verify the specificity of the peptide as a surrogate for the parent protein. Although a BLAST search could be done manually, MRMPath makes it a simple, clickable action. Indeed, the user can click just once to carry out a BLAST search on all peptides from a protein, although it may take several minutes for the BLAST search to be completed. The result of the MRMPath BLAST search may reveal that there is only one protein record that matches the peptide sequence, or it may indicate that there are multiple protein records. The latter may nonetheless be all the same protein since the NCBI database has many duplicate records for each protein. To assist the investigator, MRMPath-generated BLAST table of results contains a link to the full protein record. It should be noted that although MRMPath in combination with the BLAST search may indicate that a protein is specific in alphabetic space, the MRM-MS analysis is carried out in mass space and therefore the possibility remains of peptides with similar sequences that match the m/z of the precursor ion and the selected product ion. This would occur when a peptide had the same amino acids (i.e.,

Figure 5: (a) The user interface for MRMutation. A user can input a free text search of the Accession ID of a UniProt entry. (b) The results (truncated) of a search in the interface identified records with the keyword "p53." (c) Clicking the first link results in the creation of a tryptic digest of the protein identified through Accession ID P04637. The mutated amino acid residues are highlighted in the tryptic peptide sequence, along with the *m/z* of the parent peptide ion.

the same molecular weight), but in a different order. In that case, there is value in obtaining the whole mass spectrum of product ions to verify the identity of the peptide. Although this is not easily obtained on a triple quadrupole (qqq) mass spectrometer, high sensitivity Qq-TOF mass spectrometers can provide this information.

As biomedical science moves into more and more use of DNA deep sequencing methods based on direct sequencing rather than hybridization to "known" sequences, it is becoming apparent that there is far more sequence variation in genes and hence proteins than previously realized. For some genes, the variation in sequence can occur between tissues in the same person. While germ-line DNA information is copied faithfully from parents to their children with little error, somatic tissues can have >1500 mutations in the whole genome [7]. This suggests that the so-called canonical sequence of a gene or protein may be subject to more variation than hitherto. Because of their potential involvement in disease, certain genes/proteins have been subject to considerable attention. One of these is the human protein p53, a regulator of the G_1/S cell cycle checkpoint that is associated with many cancers.

MRMutation allows an investigator to capture known information about mutations for a specific protein. It data

mines the UniProtKB/SwissProt database, part of the EXPASY suite of programs. The investigator can either provide the protein accession number if they already know it, or they can describe the protein (e.g., "human" and "p53" would be the search terms). In the latter case, a table of proteins normally generated by the EXPASY software appears. For both, it is best to select the protein record that starts with a "P" since these records contain a compilation of all the known mutations and greatly facilitate the recovery of the required information. In the case of human p53 (P04637), there are currently (as of the preparation of this paper) mutations that lead to 1248 peptides that are different from those in wild-type human p53. At certain residue positions in this 393 amino acid protein, there are more than 10 different amino acids. For the human low-density lipoprotein receptor (P01130) there are 129 mutated peptides, whereas for NADP-dependent isocitrate dehydrogenase only two have been described. In contrast, pyruvate kinase (P30613) has 94 peptide mutations.

While there is some overlap between the utilities provided by MRMPath and Skyline software, there are also some distinct and significant differences. The principal ones are that MRMPath leverages the results of pathway analysis and is Internet browser driven. While Skyline provides detailed analysis of mass spectrometric data that is more extensive than MRMPath, its primary input is the output of a mass spectrometry experiment, namely, the DDA (data-dependent acquisition) file and therefore is not in the domain of a biologist. Skyline is capable of processing the results of and/or data from several commercial vendors. However, Skyline is available as a standalone system that only works on the Windows operating system. Furthermore, it has to be downloaded and installed. This is an advantage for those who wish to use its tools privately and offline.

From a bioinformatics and software development standpoint, the novel aspects of MRMPath and MRMutation are advancement of the notion of interoperability [8] in the realm of proteomics [9]. Interoperability is defined as the automated exchange of knowledge and data between resources that are repositories of heterogeneous (or heterogeneously stored) information. Interoperability has seen a significant rise that keeps up with the burgeoning information available online. Interoperability seeks to create a platform for information exchange while precluding the need to recreate information that is already available at the different resource.

MRMPath and MRMutation programs access the resources, EXPASY and KEGG. The software development notions employed here are innovative because it involves access to and manipulation of information available online to better serve the users of the MRM resources. Use of MRMPath and MRMutation avoids the need to transfer and store all the protein information (from EXPASY) or all the pathway protein information (from KEGG) on local servers since the stored information would have to be continually updated. In addition, use of a dynamic mode of accessing BLAST avoids the storage of a local BLAST server.

The methods discussed in this paper are easily extensible to applications in other domains [10]. In MRMPath, the web page related to a biological pathway is downloaded and the URLs of the links therein are manipulated so that MRMPath can be deployed. The KEGG pathway downloaded by the investigator is a single webpage without additional burdens being placed on the KEGG servers. All processing is done at the UAB servers. MRMPath also represents a significant boon to mass spectrometrists, who wish to obtain surrogate peptides for proteins in any pathway in the KEGG databases.

For MRMutation, if the investigator enters a UniProtKB Accession Identifier in the text field, the URL that directs the browser to the full protein record in UniProt is modified so as to extract only the FASTA formatted protein sequence of the mutant, which is then subject to further processing. On the other hand, if a protein name is entered, MRMutation will access all the UniProt protein records that include the name. The expert user must then select the appropriate record for processing by MRMutation.

The value of MRMPath and MRMutation is that they leverage existing information (without having to recreate it). There is, on the other hand, the practical matter of the use of bioinformatics-based methodologies to rapidly and efficaciously process biological knowledge (particularly knowledge that is stored remotely and heterogeneously). Performing the same manually would be overwhelming from the standpoints of efficiency, accuracy of information processed and results obtained, and the time spent.

MRMPath and MRMutation serve researchers over a range of biomedical domains. These include anybody with a proteomics-based interest in any pathway that is currently stored in KEGG. From the time an investigator enters a protein sequence into MRMPath (in one of the three ways discussed in Section 2), results will be obtained in a matter of a few seconds; the only barriers are the time taken for BLAST results. If an investigator wishes to perform MRMPath's task manually, he or she would have to access KEGG, chose a path for a species, click on the link for a specific protein, click on the FASTA formatted file for that protein, and download the FASTA file; depending on the choice of enzyme, he or she would have to create peptide fragments, filter them according to the specifications for additional processing, manually calculate y- and b-ions, take each peptide fragment and enter it into a BLAST search, and then process the final results. It is more than likely that manually performing all the steps that MRMPath would complete in a few seconds would take a few hours. If the same procedure was to be carried out for all the proteins in a pathway for a species, it would take several man-days of work, not accounting for fatigue and the consequent errors. For MRMutation, processing of proteomic data is even more efficient.

MRMPath and MRMutation are therefore an advantageous not just from the developmental standpoint, of an interoperability-based software, but also for their simplicity of use and significant advantage over manually accomplishing the same task. These software are freely accessible and need not be downloaded and installed. The only requirements are an Internet browser; hence, the systems are platform independent. All the processing takes place on the server side,

and the graphical user interface for querying the system and the results are available instantly and dynamically on the same browser MRMPath and MRMutation can be freely accessed at http://tmpl.uab.edu/MRMPath/.

4. Conclusion

In summary, MRMPath and MRMutation are bioinformatics tools that will have great value in the design of experiments in quantitative proteomics, particularly in the analysis of biomarkers.

Acknowledgments

This study was supported in part by a Grants-in-Aid (R21 AT004661, S. Barnes, PI) from the National Center for Complementary and Alternative Medicine as a supplement provided under the American Recovery and Reinvestment Act, from the National Institute on Deafness and Other Communication Disorders, National Institutes of Health (1R21DC011068, C. Crasto, PI), and from UAB Center for Clinical and Translational Science (5UL1RR025777).

References

[1] F. Hillenkamp and M. Karas, "Mass spectrometry of peptides and proteins by matrix-assisted ultraviolet laser desorption/ionization," *Methods in Enzymology*, vol. 193, pp. 280–295, 1990.

[2] J. B. Fenn, M. Mann, C. K. Meng, S. F. Wong, and C. M. Whitehouse, "Electrospray ionization for mass spectrometry of large biomolecules," *Science*, vol. 246, no. 4926, pp. 64–71, 1989.

[3] A. P. Diz, A. Carvajal-Rodríguez, and D. O. F. Skibinski, "Multiple hypothesis testing in proteomics: a strategy for experimental work," *Molecular and Cellular Proteomics*, vol. 10, no. 3, 2011.

[4] D. F. Ransohoff, "Proteomics research to discover markers: what can we learn from netflix?" *Clinical Chemistry*, vol. 56, no. 2, pp. 172–176, 2010.

[5] S. A. Gerber, J. Rush, O. Stemman, M. W. Kirschner, and S. P. Gygi, "Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 12, pp. 6940–6945, 2003.

[6] http://www.nist.gov/pml/data/comp.cfm/.

[7] D. F. Conrad, J. E. M. Keebler, M. A. Depristo et al., "Variation in genome-wide mutation rates within and between human families," *Nature Genetics*, vol. 43, no. 7, pp. 712–714, 2011.

[8] K. H. Buetow, "Cyberinfrastructure: empowering a "third way" in biomedical research," *Science*, vol. 308, no. 5723, pp. 821–824, 2005.

[9] M. Cannataro, "Computational proteomics: management and analysis of proteomics data," *Briefings in Bioinformatics*, vol. 9, no. 2, pp. 97–101, 2008.

[10] W. Litwin, L. Mark, and N. Roussopoulos, "Interoperability of multiple autonomous databases," *Computing surveys*, vol. 22, no. 3, pp. 267–293, 1990.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2012, Article ID 534810, 9 pages
doi:10.1155/2012/534810

Research Article

Intervention in Biological Phenomena via Feedback Linearization

Mohamed Amine Fnaiech,[1] Hazem Nounou,[1] Mohamed Nounou,[2] and Aniruddha Datta[3]

[1] *Electrical and Computer Engineering Program, Texas A&M University at Qatar, P.O. Box 23874, Doha, Qatar*
[2] *Chemical Engineering Program, Texas A&M University at Qatar, P.O. Box 23874, Doha, Qatar*
[3] *Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA*

Correspondence should be addressed to Hazem Nounou, hazem.nounou@qatar.tamu.edu

Received 5 July 2012; Accepted 10 October 2012

Academic Editor: Erchin Serpedin

The problems of modeling and intervention of biological phenomena have captured the interest of many researchers in the past few decades. The aim of the therapeutic intervention strategies is to move an undesirable state of a diseased network towards a more desirable one. Such an objective can be achieved by the application of drugs to act on some genes/metabolites that experience the undesirable behavior. For the purpose of design and analysis of intervention strategies, mathematical models that can capture the complex dynamics of the biological systems are needed. S-systems, which offer a good compromise between accuracy and mathematical flexibility, are a promising framework for modeling the dynamical behavior of biological phenomena. Due to the complex nonlinear dynamics of the biological phenomena represented by S-systems, nonlinear intervention schemes are needed to cope with the complexity of the nonlinear S-system models. Here, we present an intervention technique based on feedback linearization for biological phenomena modeled by S-systems. This technique is based on perfect knowledge of the S-system model. The proposed intervention technique is applied to the glycolytic-glycogenolytic pathway, and simulation results presented demonstrate the effectiveness of the proposed technique.

1. Introduction

Biological systems are complex processes with nonlinear dynamics. S-systems are proposed in [1, 2] as a canonical nonlinear model to capture the dynamical behavior of a large class of biological phenomena [3, 4]. They are characterized by a good tradeoff between accuracy and mathematical flexibility [5]. In this modeling approach, nonlinear systems are approximated by products of power-law functions which are derived from multivariate linearization in logarithmic coordinates. It has been shown that this type of representation is a valid description of biological processes in a variety of settings. S-systems have been proposed in the literature to mathematically capture the behavior of genetic regulatory networks [6–13]. Moreover, the problem of estimating the S-system model parameters, the rate coefficients and the kinetic orders, has been addressed by several researchers [12, 14–16]. In [17], the authors studied the controllability of S-systems based on feedback linearization approach.

Recently, the authors in [18] developed two different intervention strategies, namely, indirect and direct, for biological phenomena modeled by S-systems. The goal of these intervention strategies is to transfer the target variables from an initial steady-state level to a desired final one by manipulating the control variables. The complexity of the nonlinear biological models led researchers to focus on nonlinear control approaches, such as sliding mode control that was introduced in [19].

A basic problem in control theory is how to use feedback in order to modify the original internal dynamics of nonlinear systems to achieve some prescribed behavior [20]. In particular, feedback linearization can be used for the purpose of imposing, on the associated closed-loop system, a desired behavior of some prescribed autonomous linear system. When the system to be controlled is linear time-invariant system, this is known as the problem of pole placement, while in the more general case of nonlinear systems, this is known as the problem of feedback linearization [21, 22].

Significant advances have been made in the theory of nonlinear state feedback control, such as feedback linearization and input-output decoupling techniques [21, 22]. The state feedback linearization technique has been widely utilized in many applications. For example, the authors in [23] have used feedback linearization in cancer therapy, where full knowledge of the state and parameter vectors is assumed to transform a multiinput multioutput nonlinear system into a linear and controllable one using nonlinear state feedback. Then, linear control techniques can be applied for the resulting system [22, 24].

Hence, in this paper we consider the problem designing a nonlinear intervention strategy based on feedback linearization for biological phenomena modeled by S-systems. In this proposed algorithm, the control variables are designed such that an integral action is added to the system. The main advantage of the integral action is in improving the steady state performance of the closed-loop system. As a case study, the proposed intervention strategy is applied to a glycolytic-glycogenolytic pathway model. The glycolytic-glycogenolytic pathway model is selected as it plays an important role in cellular energy generation when the level of glucose in the blood is low (fasting state) and glycogen has to be broken down to provide the substrate to run glycolysis. By controlling the glycogenolytic reaction, one can exert control over whether glycolysis will run or not under low-glucose conditions.

This paper is organized as follows. In Section 2, the S-system model is presented and the control problem is formulated. In Section 3, some mathematical preliminaries as well as the feedback linearizable control scheme are presented. In Section 4, the glycolytic-glycogenolytic pathway model is considered as a case study. Finally, concluding remarks and possible future research directions are outlined in Section 5.

2. S-System Presentation and Problem Formulation

Consider the following S-system model [25]:

$$\dot{x}_i = \alpha_i \prod_{j=1}^{N+m} x_j^{\theta_{ij}} - \beta_i \prod_{j=1}^{N+m} x_j^{\mu_{ij}}, \quad i = 1, 2, \ldots, N, \tag{1}$$

where $\alpha_i > 0$ and $\beta_i > 0$ are rate coefficients and θ_{ij} and μ_{ij} are kinetic orders and there exist $N + m$ variables (genes/metabolites) where the first N variables are dependent and the remaining m variables are independent variables. Assume that p out of the N dependent variables are target (or output) variables (i.e., genes/metabolites that need to be regulated to some desired final values), where these output variables are defined as

$$y_j = x_i, \quad j = 1, \ldots, p, \tag{2}$$

and $i \in \mathcal{Y} \subset \{1, \ldots, N\}$, where $\mathcal{Y}$ is the set of indices corresponding to the dependent variables that are selected as output variables. The steady-state analysis of the S-system model [1, 18] shows that when the number of dependent variables with prespecified desired values is equal to the number of independent variables (which means that we have enough degrees of freedom), the above S-system model equations will have a unique steady-state solution under the nonsingularity assumption. Hence, in order to control the expressions/concentrations of the target variables, we consider an integral control approach where the following r equations are added to the above S-system:

$$\dot{x}_i = u_j, \quad j = 1, \ldots, p, \tag{3}$$

where $i \in \mathcal{U} \subset \{N + 1, \ldots, N + m\}$, where $\mathcal{U}$ is the set of indices corresponding to the independent variables that are used as control variables. This means that r out of the m independent variables will be used as control variables, and the overall system will have p inputs and p outputs. It should be noted that the formulation above can be easily extended to deal with systems having more inputs than outputs. Let us denote by $\mathcal{X} = \{1, \ldots, N\} + \mathcal{U}$, where $\mathcal{X}$ corresponds to the indices of all variables except the independent variables that are not used as control variables. Here, it is assumed that the values (expressions/concentrations) of the independent variables that are not used as control variables are known constants (i.e., $x_i = \delta_i$, $i \in \{N + 1, \ldots, N + m\} - \mathcal{U}$, where δ_i are known constants) [6].

Figure 1 shows the S-system (1) augmented by the integral control. The S-system with integral control (1)–(3) can be written in the form

$$\begin{aligned} \dot{x} &= f(x) + g(x)u, \\ y &= h(x), \end{aligned} \tag{4}$$

where $x = [x_i]^T \in \mathbb{R}^{N+p}$, $i \in \mathcal{X}$, $u = [u_1, \ldots, u_p]^T \in \mathbb{R}^p$, $y = [y_1, \ldots, y_p]^T \in \mathbb{R}^p$ and

$$f(x) = \left.\begin{bmatrix} \alpha_1 \prod_{j=1}^{N+m} x_j^{\theta_{1j}} - \beta_1 \prod_{j=1}^{N+m} x_j^{\mu_{1j}} \\ \vdots \\ \alpha_N \prod_{j=1}^{N+m} x_j^{\theta_{Nj}} - \beta_N \prod_{j=1}^{N+m} x_j^{\mu_{Nj}} \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right\} p, \tag{5}$$

$$g(x) = \begin{bmatrix} 0_{N\times p} \\ I_{p\times p} \end{bmatrix}, \quad h(x) = [x_i]^T, \quad i \in \mathcal{Y},$$

which can be expressed as

$$\dot{x} = f(x) + \sum_{i=1}^{p} g_i(x) u_i, \tag{6}$$

$$y_i = h_i(x), \tag{7}$$

where $g_i(x) = [0_1, 0_2, \ldots, 0_{N+i-1}, 1_{N+i}, 0_{N+i+1}, \ldots, 0_{N+p}]^T$, for $i = 1, \ldots, p$.

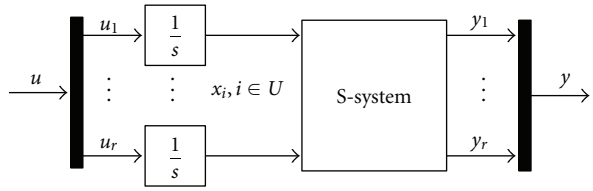


Figure 1: S-system with integral control architecture.

Problem Formulation. Suppose that the outputs of the S-system (1) are initially at the steady-state condition y_{0_j}, $j = 1,\ldots,p$. Let us denote by y_{d_j}, $j = 1,\ldots,p$, the desired final steady state values of the output (target) variables. Then, the main goal of the feedback linearizable controller is to determinate the control inputs u_j, $j = 1,\ldots,p$, that can guide the target variables from the initial steady-state condition to the final one [18].

3. Feedback Linearizable Intervention

Here, we show how feedback linearization can be utilized to design a nonlinear intervention strategy to control biological phenomena modeled by S-systems. Feedback linearization can be used to obtain a linear relationship between the output vector y and a new input vector v, by making a right choice of the linearizing law. Once the equivalent model becomes linear, we may design a dynamic control law-based classical linear control theory. Before starting the development of this control technique, it is important to introduce the following mathematical preliminaries [20–22].

3.1. Mathematical Preliminaries. Let the vector function $f : \Re^n \to \Re^n$ be a vector field in $\Re^n$. The vector function $f(x)$ is called a smooth vector function if it has continuous partial derivatives of any required order [26]. Given a scalar function $h(x)$ and a vector field $f(x)$, we define a new scalar function $L_f h$, called the Lie derivative of h with respect to f, as follows.

Definition 1 (see [26]). Let $h : \Re^n \to \Re$ be a smooth scalar function, and $f : \Re^n \to \Re^n$ be a smooth vector field on $\Re^n$, then the Lie derivative of h with respect to f is a scalar function defined by $L_f h = \nabla h f$.

Thus, the Lie derivative $L_f h$ is simply the directional derivative of h along the direction of the vector f. Repeated Lie derivatives can be defined recursively as follows:

$$L_f^{(0)} h = h,$$
$$L_f^{(i)} h = L_f\left(L_f^{(i-1)} h\right) = \nabla\left(L_f^{(i-1)} h\right) f, \quad \text{for } i = 1, 2, \ldots. \tag{8}$$

Similarly, if g is another vector field, then the scalar function $L_g L_f h(x)$ can be described as

$$L_g L_f h = \nabla\left(L_f h\right) g. \tag{9}$$

Definition 2 (see [26]). Let f and g be two vector fields on $\Re^n$. The Lie bracket of f and g is a third vector field defined by

$$[f, g] = \nabla g f - \nabla f g, \tag{10}$$

where the Lie bracket $[f, g]$ is commonly written as $\mathrm{ad}_f g$ (where ad stands for "adjoint").

Repeated Lie brackets can then be defined recursively by $\mathrm{ad}_f^{(0)} g = g, \ldots, \mathrm{ad}_f^{(i)} g = [f, \mathrm{ad}_f^{(i-1)} g]$.

3.2. Feedback Linearizable Controller. Consider the S-system model (6). Differentiating the jth output y_j of this system with respect to time, we get

$$\dot{y}_j = L_f h_j(x) + \sum_{i=1}^{p}\left(L_{g_i} h_j(x)\right) u_i, \tag{11}$$

for $j = 1, 2, 3, \ldots p$. Note in (7) that if each of the $L_{g_i} h_j(x) = 0$, then the inputs do not appear in the equation. Define γ_j to be the smallest integer such that at least one of the inputs appears in $y_j^{(\gamma_j)}$, that is

$$y_i^{(\gamma_j)} = L_f^{(\gamma_j)} h_j(x) + \sum_{i=1}^{p} L_{g_i}\left(L_f^{(\gamma_j - 1)} h_j(x)\right) u_i, \tag{12}$$

with at least one of the $L_{g_i}(L_f^{(\gamma_j-1)} h_j) \neq 0$, for some x. Let the $p \times p$ matrix $D(x)$ be defined as

$$D(x) = \begin{pmatrix} L_{g_1} L_f^{(\gamma_1 - 1)} h_1 & L_{g_2} L_f^{(\gamma_1 - 1)} h_1 & \cdots & L_{g_p} L_f^{(\gamma_1 - 1)} h_1 \\ L_{g_1} L_f^{(\gamma_2 - 1)} h_2 & L_{g_2} L_f^{(\gamma_2 - 1)} h_2 & \cdots & L_{g_p} L_f^{(\gamma_2 - 1)} h_2 \\ \vdots & \vdots & & \vdots \\ L_{g_1} L_f^{(\gamma_p - 1)} h_p & L_{g_2} L_f^{(\gamma_p - 1)} h_p & \cdots & L_{g_p} L_f^{(\gamma_p - 1)} h_p \end{pmatrix}. \tag{13}$$

Based on the above definitions, the relative degree for multi-input multioutput (MIMO) systems is defined next.

Definition 3 (see [27]). The system (6)-(7) is said to have vector relative degree $\gamma_1, \gamma_2, \ldots, \gamma_p$ at x_0 if $L_{g_i} L_f^{(k)} h_i(x) \equiv 0$, $0 \leq k \leq \gamma_i - 2$, for $i = 1, \ldots, p$ and the matrix $D(x_0)$ is nonsingular.

If a system has well-defined vector relative degree, then (12) can be expressed as

$$\left[y_1^{(\gamma_1)}, y_2^{(\gamma_2)}, \ldots, y_p^{(\gamma_p)}\right]^T = \xi(x) + D(x) u, \tag{14}$$

where

$$\xi(x) = \left[L_f^{(\gamma_1)} h_1(x), L_f^{(\gamma_2)} h_2(x), \ldots, L_f^{(\gamma_p)} h_p(x)\right]^T. \tag{15}$$

Since $D(x_0)$ is nonsingular, it follows that $D(x) \in \Re^{p \times p}$ is bounded away from nonsingularity for $x \in U$, a neighborhood U of x_0. Then, the state feedback control law

$$u = D(x)^{-1}(-\xi(x) + v) \tag{16}$$

yields the linear closed-loop system

$$y_i^{(\gamma_j)} = v_i. \tag{17}$$

The block diagram of the linearized system is shown in Figure 2.

Feedback linearization transforms the system into a linear system where linear control approaches can be applied. Here, v represents the new input vector of the linearized system.

In the case the system has vector relative degree, where $\gamma_1 + \cdots + \gamma_p = n$, the nonlinear system can be converted into a controllable linear system, where the feedback control law is defined in (16) and the coordinate transformation is $\xi(x) = [L_f^{(j)} h_i(x)]^T$, $0 \le j \le \gamma_i - 1$, $0 \le i \le p$. Let the following distributions be defined as [27]

$$\begin{aligned} G_0(x) &= \operatorname{span}\left\{g_1(x), \ldots, g_p(x)\right\}, \\ G_1(x) &= \operatorname{span}\left\{g_1(x), \ldots, g_p(x), \operatorname{ad}_f g_1, \ldots, \operatorname{ad}_f g_p(x)\right\}, \\ &\vdots \\ G_i(x) &= \operatorname{span}\left\{\operatorname{ad}_f^{(k)} g_i(x) : 0 \le k \le i,\ 0 \le j \le p\right\}, \end{aligned} \tag{18}$$

for $i = 1, \ldots, n-1$, then we have the following result.

Proposition 4 (see [27]). *Suppose that the matrix $g(x_0)$ has rank p. Then, there exist p functions $\lambda_1, \ldots, \lambda_p$, such that the system*

$$\begin{aligned} \dot{x} &= f(x) + g(x)u, \\ y &= \lambda(x), \end{aligned} \tag{19}$$

has vector relative degree $(\gamma_1, \ldots, \gamma_p)$ with $\gamma_1 + \gamma_2 + \cdots + \gamma_p = n$ if

(i) *for each $0 \le i \le n-1$ the distribution G_i has constant dimension in the neighborhood U of x_0;*

(ii) *the dimension G_{n-1} has dimension n;*

(iii) *for each $0 \le i \le n-2$ the dimension G_i is involutive.*

The proof of this proposition can be found in [27].

The new control vector $v = [v_1, \ldots, v_p]^T$ is designed based on the desired closed-loop response, which can be written as

$$v_j = y_{d_j}^{(\gamma_j)} + k_{\gamma_j - 1}\left(y_{d_j}^{(\gamma_j - 1)} - y_j^{(\gamma_j - 1)}\right) + \cdots + k_1\left(y_{d_j} - y_j\right) \tag{20}$$

for $j = 1, \ldots, p$, where $\{y_{d_j}, y_{d_j}^{(1)}, \ldots, y_{d_j}^{(\gamma_j - 1)}, y_{d_j}^{(\gamma_j)}\}$ denotes the desired reference trajectories for the outputs. The proportional gains are chosen such that the following polynomial is a Hurwitz polynomial [28]:

$$s^{\gamma_j} + k_{\gamma_j - 1} s^{\gamma_j - 1} + \cdots + k_2 s + k_1 = 0. \tag{21}$$

The block diagram of the closed-loop system in the feedback linearizable form is shown in Figure 3.

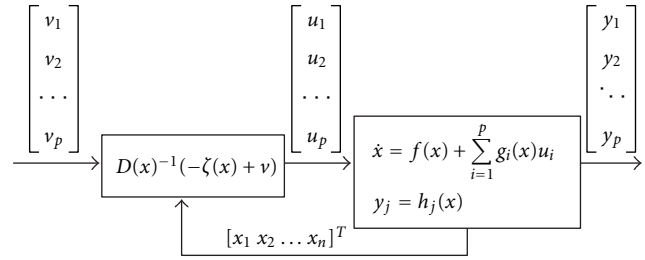


Figure 2: Diagram block of the linearizable system.

4. Case Study

In this section, we demonstrate the efficacy of the feedback linearizable intervention approach described in this paper by applying it to a well-studied biological pathway model representing the glycolytic-glycogenolytic pathway shown in Figure 4 [17, 29]. Glycolysis is the process of breaking up a six-carbon glucose molecule into two molecules of a three-carbon compound, and glycogenolysis is the process by which the stored glycogen in the body is broken up to meet the needs for glucose. In glycogenolysis, the phosphorylase enzyme acts on the polysaccharide glycogen to reduce its length by one glucose unit. The glucose unit is released as a glucose-1 phosphate. The glycolytic-glycogenolytic pathway can be mathematically represented by the following S-system model:

$$\begin{aligned} \dot{x}_1 &= \alpha_1 x_4^{\theta_{14}} x_6^{\theta_{16}} - \beta_1 x_1^{\mu_{11}} x_2^{\mu_{12}} x_7^{\mu_{17}}, \\ \dot{x}_2 &= \alpha_2 x_1^{\theta_{21}} x_2^{\theta_{22}} x_5^{\theta_{25}} x_7^{\theta_{27}} x_{10}^{\theta_{210}} - \beta_2 x_2^{\mu_{22}} x_3^{\mu_{23}} x_8^{\mu_{28}}, \\ \dot{x}_3 &= \alpha_3 x_2^{\theta_{32}} x_3^{\theta_{33}} x_8^{\theta_{38}} - \beta_3 x_3^{\mu_{33}} x_9^{\mu_{39}}. \end{aligned} \tag{22}$$

In this case, $N = 3$, $m = 7$ and the parameter are defined as $\alpha_1 = 0.077884314$, $\theta_{14} = 0.66$, $\theta_{16} = 1$, $\beta_1 = 1.06270825$, $\mu_{11} = 1.53$, $\mu_{12} = -0.59$, $\mu_{17} = 1$, $\alpha_2 = 0.585012402$, $\theta_{21} = 0.95$, $\theta_{22} = -0.41$, $\theta_{25} = 0.32$, $\theta_{27} = 0.62$, $\theta_{210} = 0.38$, $\beta_2 = \alpha_3 = 0.0007934561$, $\mu_{22} = \theta_{32} = 3.97$, $\mu_{23} = \theta_{33} = -3.06$, $\mu_{28} = \theta_{38} = 1$, $\beta_3 = 1.05880847$, $\mu_{33} = 0.3$, and $\mu_{39} = 1$. Here, the model variables are defined as follows: x_1 is glucose-1-P, x_2 is glucose-6-P, x_3 is fructose-6-P, x_4 is inorganic phosphate ion, x_5 is glucose, x_6 is phosphorylase a, x_7 is phosphoglucomutase, x_8 is phosphoglucose isomerase, x_9 is phosphofructokinase, and x_{10} is glucokinase.

For this model, the metabolites x_4 through x_{10} are defined as independent variables, which are the variables that are not affected by other variables, and the metabolites x_1 through x_3 are defined as the dependent variables, which are the primary variables of interest that we wish to control. Here, we choose the independent variables x_4, x_5, and x_8 as manipulated or control variables, as shown in Figure 4, as they can affect the production of the dependent variables x_1, x_2, and x_3. Also, we choose to keep the independent variables x_6, x_7, x_9, and x_{10} fixed ignoring their effect on the controlled variables, and assuming that the controller only uses the independent variables x_4, x_5, and x_8 to control the dependent variables x_1, x_2, and x_3. The independent variables have the following values $x_4 = 10$, $x_5 = 5$, $x_6 = 3$, $x_7 = 40$, $x_8 = 136$,

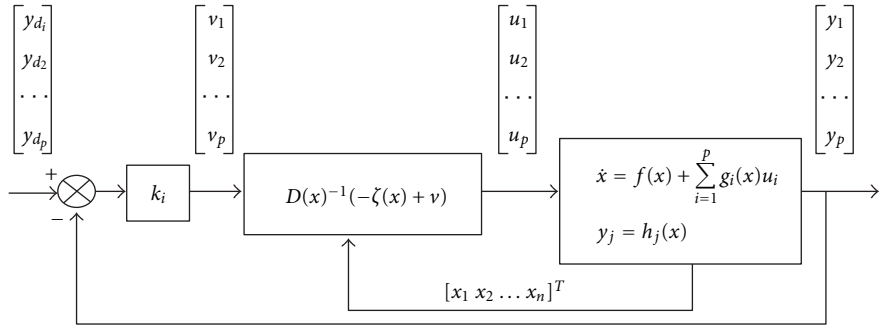


Figure 3: Closed loop of the linearizable system.

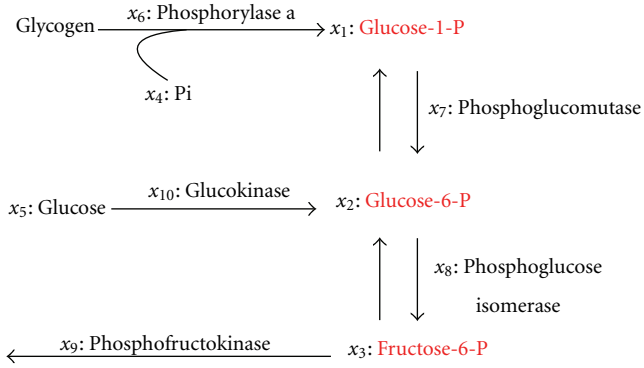


Figure 4: Glycolytic-glycogenolytic pathway [29].

$x_9 = 2.86$, and $x_{10} = 4$. Here, we try to control x_1, x_2, and x_3 by manipulating x_4, x_5, and x_8, so we have

$$\begin{aligned} y_1 &= x_1, \\ y_2 &= x_2, \\ y_3 &= x_3, \\ \dot{x}_4 &= u_1, \\ \dot{x}_5 &= u_2, \\ \dot{x}_8 &= u_3, \end{aligned} \tag{23}$$

and all other $x_i's$ for $i = 6, 7, 9$, and 10 are kept fixed. The initial values of the outputs y_1, y_2, and y_3 are selected as 0.067, 0.465, and 0.150, respectively, and the desired reference outputs are selected as $y_{d1} = 0.2$, $y_{d2} = 0.5$, and $y_{d3} = 0.4$.

Hence, the overall system can be expressed in the form of (6), where

$$f(x) = \begin{bmatrix} a_1 x_4^{\theta_{14}} - b_1 x_1^{\mu_{11}} x_2^{\mu_{12}} \\ a_2 x_1^{\theta_{21}} x_2^{\theta_{22}} x_5^{\theta_{25}} - b_2 x_2^{\mu_{22}} x_3^{\mu_{23}} x_8^{\mu_{28}} \\ a_3 x_2^{\theta_{32}} x_3^{\theta_{33}} x_8^{\theta_{38}} - b_3 x_3^{\mu_{33}} \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

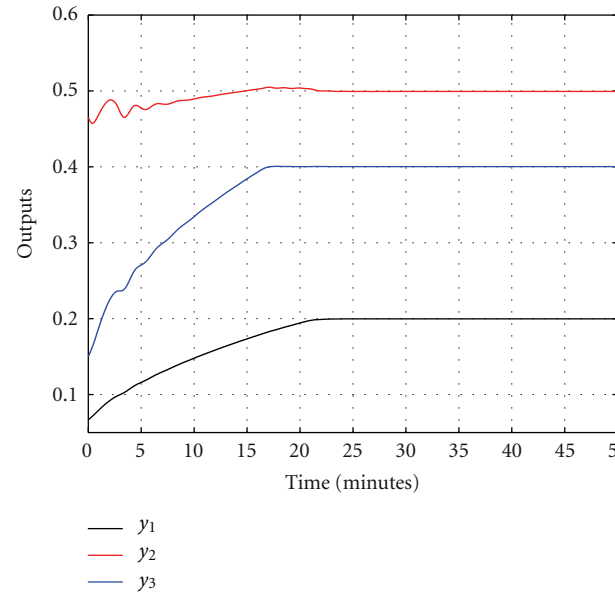


Figure 5: Closed-loop outputs for constant reference signals.

$$\begin{aligned} g(x) &= [g_1(x), g_2(x), g_3(x)] \\ &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T, \\ h(x) &= \begin{bmatrix} h_1(x) \\ h_2(x) \\ h_3(x) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \end{aligned} \tag{24}$$

where $a_1 = \alpha_1 x_6^{\theta_{16}}$, $a_2 = \alpha_2 x_7^{\theta_{27}} x_{10}^{\theta_{210}}$, $a_3 = \alpha_3$, $b_1 = \beta_1 x_7^{\mu_{17}}$, $b_2 = \beta_2$, and $b_3 = \beta_3 x_9^{\mu_{39}}$.

Based on the S-system model describing the glycolytic-glycogenolytic pathway, it can be verified that the outputs need to be differentiated twice with respect to time so that the input variables (u_1, u_2, or u_3) appear in the expressions of differentiated outputs, as follows:

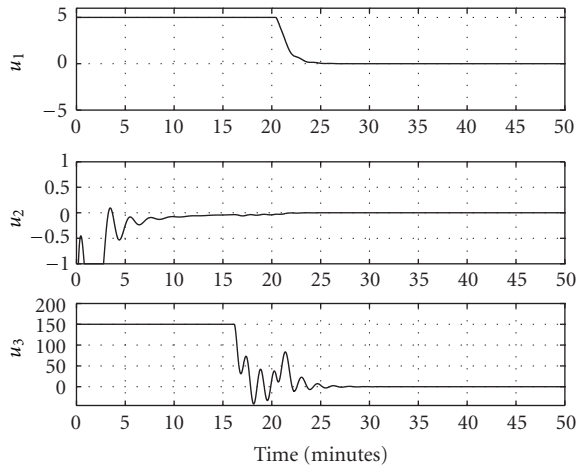


Figure 6: Control signals for constant reference signals.

$$
\begin{aligned}
y_1^{(2)} &= L_f^{(2)}h_1(x) + L_{g_1}\left(L_f h_1(x)\right)u_1,\\
y_2^{(2)} &= L_f^{(2)}h_2(x) + L_{g_2}\left(L_f h_2(x)\right)u_2 + L_{g_3}\left(L_f h_2(x)\right)u_3,\\
y_3^{(2)} &= L_f^{(2)}h_3(x) + L_{g_3}\left(L_f h_3(x)\right)u_3,
\end{aligned}
\tag{25}
$$

where

$$
\begin{aligned}
L_f^{(2)}h_1(x) &= b_1^2\mu_{11}x_1^{2\mu_{11}-1}x_2^{2\mu_{12}}\\
&\quad - b_1a_1\mu_{11}x_1^{\mu_{11}-1}x_2^{\mu_{12}}x_4^{\theta_{14}}\\
&\quad + b_1b_2\mu_{12}x_2^{\mu_{12}+\mu_{22}-1}x_1^{\mu_{11}}x_3^{\mu_{23}}x_8^{\mu_{28}}\\
&\quad - b_1a_2\mu_{12}x_2^{\mu_{12}+\theta_{21}-1}x_1^{\mu_{11}+\theta_{21}}x_5^{\theta_{25}},\\
L_f^{(2)}\mu_2(x) &= a_1a_2\theta_{21}x_1^{\theta_{21}-1}x_2^{\theta_{22}}x_4^{\theta_{14}}x_5^{\theta_{25}}\\
&\quad - a_2b_1x_1^{\theta_{21}+\mu_{11}-1}x_2^{\theta_{22}+\mu_{12}}x_5^{\theta_{25}}\\
&\quad + a_2^2\theta_{22}x_2^{2\theta_{22}-1}x_1^{2\theta_{21}}x_5^{\theta_{25}}\\
&\quad - a_2b_2x_2^{\theta_{22}+\mu_{22}-1}x_1^{\theta_{21}}x_3^{\mu_{23}}x_5^{\theta_{25}}x_8^{\mu_{28}}\\
&\quad - b_2a_2\mu_{22}x_1^{\theta_{12}}x_2^{\mu_{22}+\theta_{22}-1}x_3^{\mu_{23}}x_5^{\theta_{25}}x_8^{\mu_{28}}\\
&\quad + b_2^2\mu_{22}x_2^{2\mu_{22}-1}x_3^{2\mu_{23}}x_8^{2\mu_{28}}\\
&\quad - b_2a_3\mu_{23}x_2^{\mu_{22}+\theta_{32}}x_3^{\mu_{23}+\theta_{33}-1}x_8^{\mu_{28}+\theta_{38}}\\
&\quad + b_2b_3\mu_{23}x_2^{\mu_{22}}x_3^{\mu_{23}+\mu_{33}-1}x_8^{\mu_{28}},\\
L_f^{(2)}\mu_3(x) &= a_3a_2\theta_{32}x_1^{\theta_{21}}x_2^{\theta_{32}+\theta_{22}-1}x_3^{\theta_{33}}x_5^{\theta_{25}}x_8^{\theta_{38}}\\
&\quad - a_3b_2\theta_{32}x_2^{\theta_{32}+\mu_{22}-1}x_3^{\theta_{33}+\mu_{23}}x_8^{\theta_{38}+\mu_{28}}\\
&\quad + a_3^2\theta_{33}x_2^{2\theta_{32}}x_3^{2\theta_{33}-1}x_8^{2\theta_{38}}\\
&\quad - a_3b_3\theta_{33}x_2^{\theta_{32}}x_3^{\theta_{33}+\mu_{33}-1}x_8^{\theta_{38}}\\
&\quad - b_3a_3\mu_{33}x_2^{\theta_{32}}x_3^{\mu_{33}+\theta_{33}-1}x_8^{\theta_{38}}\\
&\quad + b_3^2\mu_{33}x_3^{2\mu_{33}-1},
\end{aligned}
$$

$$
\begin{aligned}
L_{g_1}\left(L_f h_1(x)\right) &= a_1x_4^{\theta_{14}-1},\\
L_{g_2}\left(L_f h_2(x)\right) &= a_2\theta_{25}x_1^{\theta_{21}}x_2^{\theta_{22}}x_5^{\theta_{25}-1},\\
L_{g_3}\left(L_f h_2(x)\right) &= -b_2\mu_{28}x_2^{\mu_{22}}x_3^{\mu_{23}}x_8^{\mu_{28}-1},\\
L_{g_3}\left(L_f h_3(x)\right) &= a_3\theta_{38}x_2^{\theta_{32}}x_3^{\theta_{33}}x_8^{\theta_{38}-1}.
\end{aligned}
\tag{26}
$$

Hence, in this case the system has vector relative degree $\gamma = [\gamma_1, \gamma_2, \gamma_3]^T = [2, 2, 2]^T$, and hence we have $\gamma_1 + \gamma_2 + \gamma_3 = 6$.

The matrix form of the system of differential equations presented in (25) can be written in the form of (14), where

$$
\xi(x) = \left[L_f^{(2)}h_1(x), L_f^{(2)}h_2(x), L_f^{(2)}h_3(x)\right]^T,
$$

$$
D(x) = \begin{pmatrix}
L_{g_1}\left(L_f h_1(x)\right) & 0 & 0\\
0 & L_{g_2}\left(L_f h_2(x)\right) & L_{g_3}\left(L_f h_2(x)\right)\\
0 & 0 & L_{g_3}\left(L_f h_3(x)\right)
\end{pmatrix}.
\tag{27}
$$

The matrix $D(x)$ is invertible if the following condition is satisfied:

$$
L_{g_1}\left(L_f h_1(x)\right) \times L_{g_2}\left(L_f h_2(x)\right) \times L_{g_3}\left(L_f h_3(x)\right) \neq 0. \tag{28}
$$

Based on (25), it can be seen that the control variables u_1 and u_2 appear only in the expressions of $y_1^{(2)}$ and $y_2^{(2)}$, respectively. However, u_3 appears in the expressions of $y_2^{(2)}$ and $y_3^{(2)}$. Hence, u_1 and u_3 need to be used to control y_1 and y_3, respectively, and both u_2 and u_3 are needed to control y_2.

Hence, the control laws based on (16) can be expressed as

$$
\begin{aligned}
u_1 &= \frac{\left(-L_f^{(2)}h_1(x) + \nu_1\right)}{L_{g_1}\left(L_f h_1(x)\right)},\\
u_2 &= \frac{\left(-L_f^{(2)}h_2(x) - L_{g_3}\left(L_f h_3(x)\right)u_3 + \nu_2\right)}{L_{g_2}\left(L_f h_2(x)\right)},\\
u_3 &= \frac{\left(-L_f^{(2)}h_3(x) + \nu_3\right)}{L_{g_3}\left(L_f h_3(x)\right)}.
\end{aligned}
\tag{29}
$$

Substituting the expressions of the control variables (29) in (25), we obtain the following decoupled linear system:

$$
\begin{aligned}
y_1^{(2)} &= \nu_1,\\
y_2^{(2)} &= \nu_2,\\
y_3^{(2)} &= \nu_3.
\end{aligned}
\tag{30}
$$

The new control variables ν_j, for $j = 1, 2, 3$, need to be designed so that the target variables y_j track some desired reference trajectories, y_{d_j}.

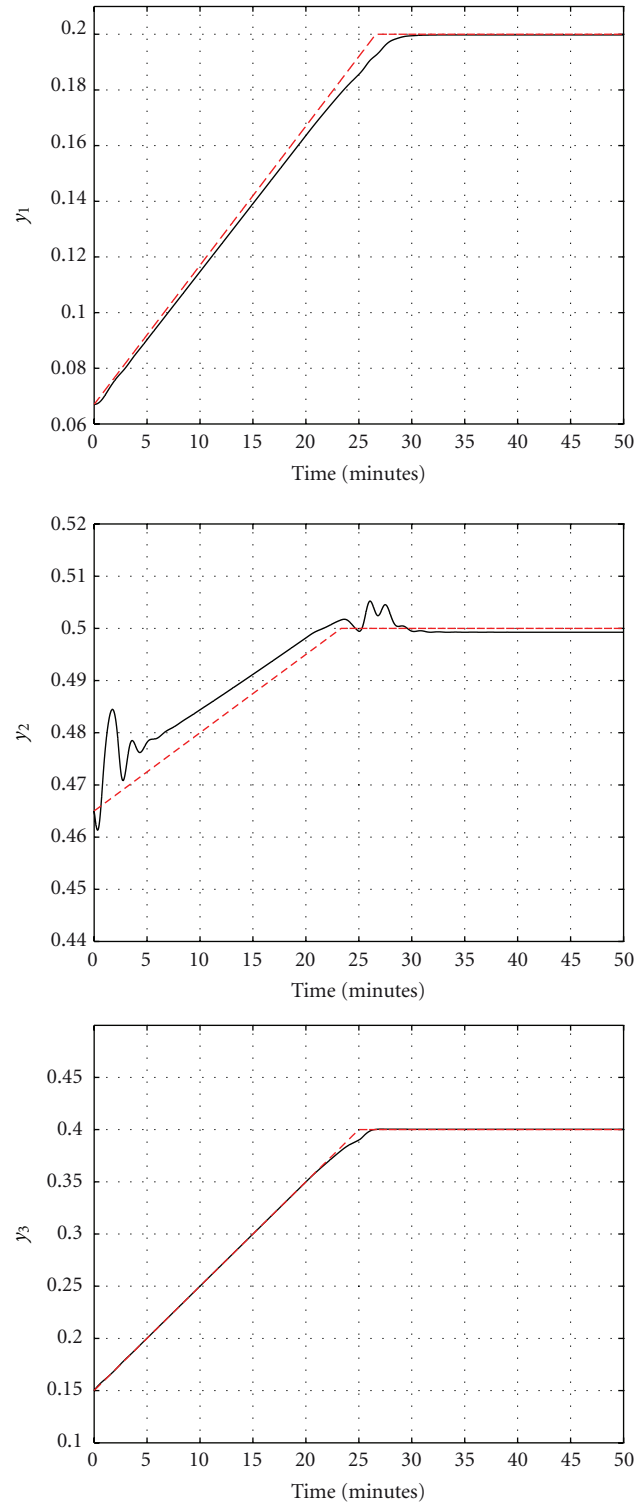


Figure 7: Output response for closed-loop tracking.

Using (20), the new control variables v_j, for $j = 1, 2, 3$, are found to be

$$\begin{aligned} v_1 &= \ddot{y}_{d_1} + k_1(\dot{y}_{d_1} - \dot{y}_1) + k_{11}(y_{d_1} - y_1), \\ v_2 &= \ddot{y}_{d_2} + k_2(\dot{y}_{d_2} - \dot{y}_2) + k_{21}(y_{d_2} - y_2), \\ v_3 &= \ddot{y}_{d_3} + k_3(\dot{y}_{d_3} - \dot{y}_3) + k_{31}(y_{d_3} - y_3). \end{aligned} \tag{31}$$

The new control components, v_1, v_2, and v_3, are defined in (31), where the parameters are selected as $k_1 = 1$, $k_{11} = 5$, $k_2 = 10^{-3}$, $k_{21} = 20$, $k_3 = 3$, and $k_{31} = 5$.

Figures 5 and 6 show the output response and the control input signals when the feedback linearizable controller is applied. It is clear from Figure 5 that the system outputs

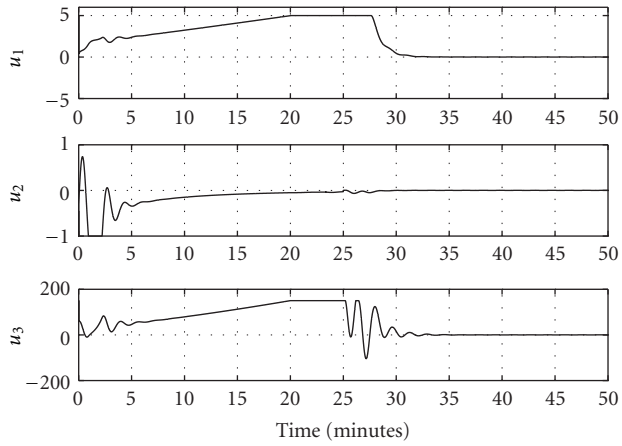


Figure 8: Control signals for closed-loop tracking.

converge to their desired values. Another simulation study is implemented for a different reference trajectory, where the value of the reference signal increases linearly before saturating at the desired final value. The closed-loop output response in this case is shown in Figure 7 and the control signals are shown in Figure 8. It is clear from Figure 7 that the feedback linearizable controller is driving the target variables to track the desired reference trajectories.

To study the robustness properties of the feedback linearizable controller, similar simulation studies have been conducted when the parameters μ_{22} and β_2 are varied within 10% of their nominal values. It has been found that the closed-loop system is stable only for parameter variations within 1% and with unacceptable performance. This agrees with our earlier assumption that full system knowledge is needed for proper operation of the feedback linearizable controller.

5. Conclusion

In this paper, feedback linearizable control has been applied for intervention of biological phenomena modeled in the S-system framework. As a case study, the glycogenolytic-glycolytic pathway model has been used to demonstrate the efficacy of feedback linearization in controlling biological phenomena modeled by S-system. One main drawback of this approach is that it assumes full knowledge of the biological system model. Usually, the S-system model does not perfectly represent the actual dynamics of the biological phenomena. Hence, one future research direction is to develop an adaptive intervention strategy that is capable of controlling the biological system even in the presence of model uncertainties. Another future research direction is to develop intervention techniques that take into account additional constraints due to the nature of the drug injection process. Definitely, incorporating such knowledge from medical practitioners would require imposing constraints on the magnitude, duration, and possibly the rate of change of the injected drug into the design of intervention technique.

Acknowledgments

This work was made possible by NPRP Grant NPRP08-148-3-051 from the Qatar National Research Fund (a Member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] M. A. Savageau, "Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions," *Journal of Theoretical Biology*, vol. 25, no. 3, pp. 365–369, 1969.

[2] E. O. Voit, *Canonical Nonlinear Modeling: S-System Approach to Understanding Complexity*, Van Nostrand/Reinhold, New York, NY, USA, 1991.

[3] E. O. Voit, "A systems-theoretical framework for health and disease: inflammation and preconditioning from an abstract modeling point of view," *Mathematical Biosciences*, vol. 217, no. 1, pp. 11–18, 2009.

[4] E. O. Voit, F. Alvarez-Vasquez, and Y. A. Hannun, "Computational analysis of sphingolipid pathway systems," *Advances in Experimental Medicine and Biology*, vol. 688, pp. 264–275, 2010.

[5] R. Gentilini, "Toward integration of systems biology formalism: the gene regulatory networks case," *Genome informatics. International Conference on Genome Informatics.*, vol. 16, no. 2, pp. 215–224, 2005.

[6] E. O. Voit and J. Almeida, "Decoupling dynamical systems for pathway identification from metabolic profiles," *Bioinformatics*, vol. 20, no. 11, pp. 1670–1681, 2004.

[7] I. C. Chou, H. Martens, and E. O. Voit, "Parameter estimation in biochemical systems models with alternating regression," *Theoretical Biology and Medical Modelling*, vol. 3, article 25, 2006.

[8] T. Kitayama, A. Kinoshita, M. Sugimoto, Y. Nakayama, and M. Tomita, "A simplified method for power-law modelling of metabolic pathways from time-course data and steady-state flux profiles," *Theoretical Biology and Medical Modelling*, vol. 3, article 24, 2006.

[9] L. Qian and H. Wang, "Inference of genetic regulatory networks by evolutionary algorithm and H_∞ filtering," in *Proceedings of the IEEE/SP 14th WorkShoP on Statistical Signal Processing (SSP '07)*, pp. 21–25, August 2007.

[10] J. Vera, R. Curto, M. Cascante, and N. V. Torres, "Detection of potential enzyme targets by metabolic modelling and optimization: application to a simple enzymopathy," *Bioinformatics*, vol. 23, no. 17, pp. 2281–2289, 2007.

[11] H. Wang, L. Qian, and E. R. Dougherty, "Steady-state analysis of genetic regulatory networks modeled by nonlinear ordinary differential equations," in *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB '09)*, pp. 182–185, April 2009.

[12] H. Wang, L. Qian, and E. Dougherty, "Inference of gene regulatory networks using S-system: a unified approach," *IET Systems Biology*, vol. 4, no. 2, pp. 145–156, 2010.

[13] A. Marin-Sanguino, S. K. Gupta, E. O. Voit, and J. Vera, "Biochemical pathway modeling tools for drug target detection in cancer and other complex diseases," *Methods in Enzymology*, vol. 487, pp. 319–369, 2011.

[14] O. R. Gonzalez, C. Küper, K. Jung, P. C. Naval, and E. Mendoza, "Parameter estimation using simulated annealing

for S-system models of biochemical networks," *Bioinformatics*, vol. 23, no. 4, pp. 480–486, 2007.

[15] Z. Kutalik, W. Tucker, and V. Moulton, "S-system parameter estimation for noisy metabolic profiles using Newton-flow analysis," *IET Systems Biology*, vol. 1, no. 3, pp. 174–180, 2007.

[16] I. C. Chou and E. O. Voit, "Recent developments in parameter estimation and structure identification of biochemical and genomic systems," *Mathematical Biosciences*, vol. 219, no. 2, pp. 57–83, 2009.

[17] A. Ervadi-Radhakrishnan and E. O. Voit, "Controllability of non-linear biochemical systems," *Mathematical Biosciences*, vol. 196, no. 1, pp. 99–123, 2005.

[18] N. Meskin, H. N. Nounou, M. Nounou, A. Datta, and E. R. Dougherty, "Intervention in biological phenomena modeled by S-systems," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 5, pp. 1260–1267, 2011.

[19] A. G. Hernández, L. Fridman, A. Levant, Y. Shtessel, S. I. Andrade, and C. R. Monsalve, "High order sliding mode controller for blood glucose in type 1 diabetes, with relative degree fluctuations," in *Proceedings of the 11th International Workshop on Variable Structure Systems (VSS '10)*, pp. 416–421, June 2010.

[20] A. Isidori, A. J. Krener, C. Gori-Giorgi, and S. Monaco, "Nonlinear decoupling via feedback: a differential geometric approach," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 331–345, 1981.

[21] A. Isidori, *Nonlinear Control Systems*, Springer, 1989.

[22] A. Isidori and M. D. Benedetto, *Feedback Linearization of Nonlinear Systems*, Taylor Francis Group-CRC Press, 2010.

[23] T. L. Chien, C. C. Chen, and C. J. Huang, "Feedback linearization control and its application to MIMO cancer immunotherapy," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 953–961, 2010.

[24] B. Jakubczyk and W. Respondek, "On linearization of control systems," *Bulletin de l'Académie Polonaise des Sciences, Série des Sciences Mathématiques*, vol. 28, pp. 517–522, 1980.

[25] E. O. Voit, *Computational Analysis of Biochemical Systems. A Practical Guide for Biochemists and Molecular Biologists*, Cambridge University Press, 2000.

[26] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*, Pearson Education, 1991.

[27] S. Sastry, *Nonlinear Systems: Analysis, Stability and Control*, Springer, 1999.

[28] R. Marino and P. Tomei, "Global adaptive output-feedback control of nonlinear systems, part II. Nonlinear parameterization," *IEEE Transactions on Automatic Control*, vol. 38, no. 1, pp. 33–48, 1993.

[29] N. V. Torres, "Modelization and experimental studies on the control of the glycolytic- glycogenolytic pathway in rat liver," *Molecular and Cellular Biochemistry*, vol. 132, no. 2, pp. 117–126, 1994.