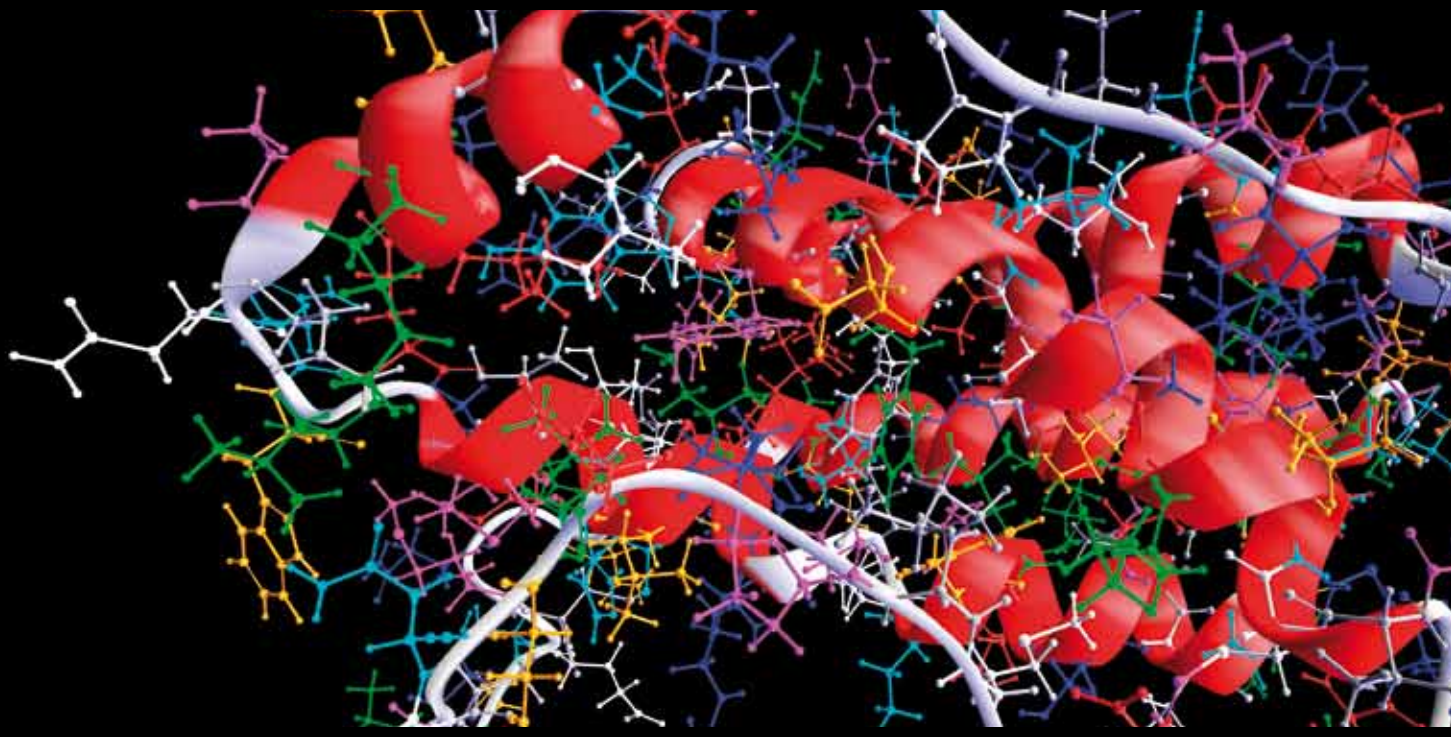
Computational and Mathematical Methods in Medicine

Applications of Machine Learning in Genomics and Systems Biology

Guest Editors: Chunmei Liu, Dongsheng Che, Xumin Liu, and Yinglei Song



Applications of Machine Learning in Genomics and Systems Biology

Computational and Mathematical Methods in Medicine

Applications of Machine Learning in Genomics and Systems Biology

Guest Editors: Chunmei Liu, Dongsheng Che, Xumin Liu, and Yinglei Song

Editorial Board

Contents

Applications of Machine Learning in Genomics and Systems Biology, Chunmei Liu, Dongsheng Che, Xumin Liu, and Yinglei Song
Volume 2013, Article ID 587492, 1 page

Efficient Identification of Transcription Factor Binding Sites with a Graph Theoretic Approach, Jia Song, Li Xu, and Hong Sun
Volume 2013, Article ID 856281, 6 pages

A Dynamic Data-Driven Framework for Biological Data Using 2D Barcodes, Hui Li and Chunmei Liu
Volume 2012, Article ID 892098, 5 pages

Biomarker Identification Using Text Mining, Hui Li and Chunmei Liu
Volume 2012, Article ID 135780, 4 pages

Identification of Novel Type III Effectors Using Latent Dirichlet Allocation, Yang Yang
Volume 2012, Article ID 696190, 6 pages

Prediction of Breeding Values for Dairy Cattle Using Artificial Neural Networks and Neuro-Fuzzy Systems, Saleh Shahinfar, Hassan Mehrabani-Yeganeh, Caro Lucas, Ahmad Kalhor, Majid Kazemian, and Kent A. Weigel
Volume 2012, Article ID 127130, 9 pages

A Novel Weighted Support Vector Machine Based on Particle Swarm Optimization for Gene Selection and Tumor Classification, Mohammad Javad Abdi, Seyed Mohammad Hosseini, and Mansoor Rezghi
Volume 2012, Article ID 320698, 7 pages

Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2013, Article ID 587492, 1 page
http://dx.doi.org/10.1155/2013/587492

Editorial

Applications of Machine Learning in Genomics and Systems Biology

Chunmei Liu,[1] Dongsheng Che,[2] Xumin Liu,[3] and Yinglei Song[4]

[1] *Department of Systems and Computer Science, Howard University, Washington, DC 20059, USA*
[2] *Department of Computer Science, East Stroudsburg University, East Stroudsburg, PA 18301, USA*
[3] *Department of Computer Science, Rochester Institute of Technology, Rochester, NY 14623, USA*
[4] *Department of Mathematics and Computer Science, University of Maryland Princess Anne, Eastern Shore, MD 21853, USA*

Correspondence should be addressed to Chunmei Liu; chunmei@scs.howard.edu

Received 15 January 2013; Accepted 15 January 2013

As the accomplishment of the human genome project, techniques that can analyze large amounts of data are urgently needed. Advances in computational techniques for analyzing high-throughput data in genomics, proteomics, and visualization have been extensively studied and have played vital roles in understanding biological mechanisms. Machine learning and related techniques such as support vector machines, Markov models, decision trees, and neural networks have been increasingly used to solve problems in genomics and systems biology.

Machine learning was defined as a "computer program that can learn from experience with respect to some class of tasks and performance measure" [1]. If we can design machine learning algorithms to learn from past experience and thus improve the performance automatically, we can solve complicated problems such as those in genomics and systems biology.

In this special issue, we have explored the topics of identifying biomarkers, transcription factor binding, novel type III effectors, predicting breeding values for dairy cattle, and gene selection and tumor classification. The papers in this volume have studied the previously researched domains and also researched the new approaches for bioinformatics problems. The papers reflect the urgency of using machine learning techniques to develop more efficient and accurate algorithms for biological problems. We hope that the papers in the volume can broaden the view of the current machine learning approaches in genomics systems biology and inspire ideas of designing new approaches for existing biological problems.

Chunmei Liu
Dongsheng Che
Xumin Liu
Yinglei Song

References

[1] T. M. Mitchell, *Machine Learning*, McGraw-Hill International, Singapore, 1997.

Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2013, Article ID 856281, 6 pages
http://dx.doi.org/10.1155/2013/856281

Research Article

Efficient Identification of Transcription Factor Binding Sites with a Graph Theoretic Approach

Jia Song,[1,2] Li Xu,[1] and Hong Sun[2]

[1] *College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*
[2] *Department of Electronic and Information Technology, Suzhou Vocational University, Suzhou 215104, China*

Correspondence should be addressed to Jia Song; yigibird@yahoo.com.cn

Received 9 October 2012; Accepted 13 December 2012

Academic Editor: Yinglei Song

Identifying transcription factor binding sites with experimental methods is often expensive and time consuming. Although many computational approaches and tools have been developed for this problem, the prediction accuracy is not satisfactory. In this paper, we develop a new computational approach that can model the relationships among all short sequence segments in the promoter regions with a graph theoretic model. Based on this model, finding the locations of transcription factor binding site is reduced to computing maximum weighted cliques in a graph with weighted edges. We have implemented this approach and used it to predict the binding sites in two organisms, *Caenorhabditis elegans* and *mus musculus*. We compared the prediction accuracy with that of the Gibbs Motif Sampler. We found that the accuracy of our approach is higher than or comparable with that of the Gibbs Motif Sampler for most of tested data and can accurately identify binding sites in cases where the Gibbs Motif Sampler has difficulty to predict their locations.

1. Introduction

Gene regulation is one of the most important biological processes at molecular level. Recent work [1] has shown that gene regulations precisely control the expression levels of genes, which in turn controls many biological processes. In general, the regulation is performed by a binding process, where a protein called *transcription factor* binds to the promoter region of a gene. The nucleotides where the binding occurs form a *binding site*. Methods that can accurately identify the locations of binding sites in the promoter regions of genes are thus crucial for understanding the process of gene regulation.

Traditionally, TF binding sites have been characterized by a variety of different experimental approaches [2, 3]. However, predicting transcription factor binding sites with experimental approaches is often expensive and time consuming. Recently, with the large amount of sequence data, computational approaches have become an important alternative to predicting the transcription factor binding sites [4–9]. Most of the available computational approaches process the promoter regions of a set of homologous genes and recognize binding sites by identifying subsequences that are similar in sequence content. Most of these approaches employ a randomized sampling procedure to identify these subsequences and thus cannot guarantee the prediction accuracy.

For example, approaches based on Gibbs sampling randomly select a candidate subsequence of a fixed length from each sequence. A sequence is arbitrarily selected and each subsequence of the same length in the sequence is aligned to a profiling model obtained from the subsequences selected on other sequences, and each subsequence in the selected sequence is thus associated with a probability value which is the alignment score. One of the subsequences is then selected based on the distribution of the probability values of these subsequences, and a new set of subsequences is thus obtained. The procedure can be repeated until the maximum allowed number of iterations has been reached or a set of satisfying local optimal subsequences have been found [4, 5, 10]. Consensus used a greedy algorithm to align functionally related sequences and applied the algorithm to identify the binding sites for the *E. coli* CRP protein [11].

Bailey and Elkan [12, 13] used the Expectation maximization technique to fit a two-component mixture model to find binding sites and developed a software MEME+. MEME+ performed better than their previous MEME software [13]. However, its accuracy for identifying transcription factor binding sites is far from being satisfactory.

Genetic algorithms simulate the process of Darwin evolutionary process to find a local optimal solution for an optimization problem. Approaches based on GAs start with an initial population of a certain size. Each individual in the population is a valid solution for the problem. The individuals in the population then go through selection, crossover, and mutation based on certain methods and probabilities to evolve to the next generation. This evolutionary procedure keeps going until the maximum allowed number of generations is reached or the difference between the results falls into a small threshold. Genetic algorithms have been applied to predict the transcription factor binding sites such as FMGA [14]. FMGA was declared to have better performance than Gibbs Motif Sampler [5] in both accuracy and computation time. MDGA [15] is another program that used genetic algorithms to find motifs in homologous sequences. It used information content to evaluate the fitness of an individual during the evolution. MDGA achieved higher accuracy than Gibbs sampling algorithm based approaches while requiring less computation time [15].

In this paper, we develop a new approach that can predict the transcription factor binding sites without using a sampling procedure to select subsequences. Our approach uses a graph to model all subsequences in the promoter regions of the homologous genes and the similarity between any pair of subsequences that are from different promoter regions. In particular, each subsequence is represented by a vertex in the graph, and two vertices are joined by an edge if the two corresponding subsequences are from different promoter regions and their similarity is higher than a threshold. The threshold can be determined using the base compositions of the promoter regions and is guaranteed to be statistically significant. Each edge in the graph is associated with a weight value which is the similarity of the two corresponding subsequences. We then compute the maximum weighted clique in the graph, and the subsequences represented by the vertices in the clique are the transcription factor binding sites.

In order to efficiently compute the maximum weighted clique in the graph, we developed an iterative approach to preprocess the vertices in the graph and remove the vertices that cannot be in the clique from the graph, the size of the graph can thus be significantly reduced by applying this technique to it. After the preprocessing stage, the algorithm exhaustively enumerates all cliques that contain as many vertices as the number of promoter region sequences in the data set and returns the one with the largest weight value.

We have implemented this approach in a computer program with C++ programming language and used the program to predict the binding sites for two organisms, *Caenorhabditis elegans* and *mus musculus*. We evaluated the prediction accuracy of our approach based on the testing data sets and compared it with that of the Gibbs Motif Sampler [5]. Our testing results showed that our approach can achieve an average developer score of 0.90 on the test data, which is higher than or comparable with that of the Gibbs Sampler. In addition, our approach can accurately identify the binding sites in some cases where the Gibbs Motif Sampler has difficulty to locate the binding sites, which is one important advantage of our approach over the Gibbs Motif Sampler.

Figure 1: Shows the procedure used to break a sequence into its subsequences.

2. Method

2.1. Notations and Problem Description. Given a graph $G = (V, E)$, a *clique* in G is a vertex subset C such that every pair of vertices in C is joined by an edge in G. If each edge in G is associated with a weight value, a *maximum weighted clique* is a clique where the sum of the weight values of all edges in the clique is the maximum of all cliques in G. The *degree* of a vertex is the number of vertices that are adjacent to it in G.

Given the sequences of the promoter regions of a set of k homologous genes, $S_1, S_2, \ldots, S_k$, the goal of the problem. The goal of the problem is to select a subsequence of a given length L from each sequence such that the sum of the similarities between all pairs of selected sequences is maximized. To compare two subsequences and evaluate their similarity, we perform a pair wise alignment between them, and the alignment score is used as a measure of the similarity. We show later in the paper that this problem can be solved by computing the maximum weighted clique in a graph.

2.2. The Algorithm. Given a set of DNA sequences $S_1, S_2, \ldots, S_k$ with similar transcription factor binding sites, we take each S_i where $i \in [1, K]$ and divide it into N-L subsequences of length L as described in Figure 1, where N is the number of nucleotides in S_i. In particular, we choose the subsequence that starts from each nucleotide in S_i and contains L nucleotides. It is not difficult to see that the number of such subsequences is N-L.

We then construct a graph G such that each vertex of the graph represents a subsequence. Subsequences from the same sequence are placed in one *column,* and all vertices in G can thus be partitioned into k disjoint columns. A sample of randomly generated and normally distributed sequence comparison scores is used to calculate a threshold which is then used to determine which sequences are similar. The algorithm starts with the first column and selects a vertex in that column and aligns its corresponding subsequence to that of every other vertex that is from a different column. If the alignment score of two subsequences is higher than the threshold, we make their vertices adjacent in G. The

Table 1: Shows the matrix used to compute comparison scores.

	A	C	G	T
A	100	−123	−28	−109
C	−123	91	−140	−28
G	−28	−140	91	−123
T	−109	−28	−123	100

algorithm repeats the above process for each vertex until all vertices in the graph have been processed.

After all the edges have been added to the graph, we proceed to preprocess the graph and remove the vertices that cannot be in a clique of size k. In particular, we examine the degree of each vertex, and if the degree of a vertex is less than k, we remove it from G. This procedure is applied iteratively to all vertices in the graph until the size of the graph cannot be further reduced.

The algorithm then starts enumerating all k-cliques in the graph and computes the weight of each clique. To this end, the algorithm assigns an integer id between 1 and k to each column in the graph and starts with columns 1 and 2. In particular, all edges that connect a vertex from column 1 and a vertex from column 2 are included in a set S. S is maintained by the algorithm to store cliques. Initially, S contains a set of edges, which are in fact cliques on two vertices. The algorithm then proceeds to examine vertices in columns 3 through k. After the algorithm completes processing column j, S contains a set of cliques of size j in G. For every vertex u in column $j + 1$, the algorithm checks each j-clique M in S to examine whether u and M can together form a $j + 1$-clique. In other words, the algorithm examines whether u is adjacent to every vertex in M or not. If it is the case, u and M are combined into a single $j + 1$-clique and included in S. After every vertex in column $j + 1$ has been processed, the algorithm examines the cliques in S and removes all j-cliques from S. It is not difficult to see that after all k columns have been processed, S contains a set of k-cliques in G. It then computes the weight value of each clique in S by adding up the weight values of all edges in the clique and outputs the clique with the largest weight value.

The matrix used to generate the comparison score is a log odds ratio matrix for comparing DNA nucleotide sequences. This matrix was developed by Chiromante et al. [16]. They followed a common approach used in protein alignment and determined substitution score by using a set of trusted aligned symbol pairs and using log odds ratio [16]. The matrix used for the alignment of short subsequences is shown in the Table 1.

In order to compute the threshold of similarity value that is used to construct the edges in G, we first compute the base composition of all promoter region sequences. Based on the percentage of each nucleotide in the base composition, we randomly generate two subsequences, each of which contains L nucleotides. We then perform a pair wise alignment between the two generated subsequences. The alignment can generate an alignment score. We repeat the procedure for a sufficiently large number of times, and we thus can obtain a large collection of alignment scores. This collection of alignment scores in fact describes the distribution of alignment scores between two subsequences from the given base composition. We then choose a confidence value p and find the smallest value c such that the percentage of the alignment scores higher than c in the collection is not larger than p. The value of c is then used as the threshold of similarity to construct graph G.

2.3. Time Complexity. Each iterative step in the preprocessing stage of the algorithm may need up to $O(N^2)$ computation time, where N is the number of nucleotides in each promoter region sequence. Since each iterative step removes at least one vertex from the graph. The preprocessing stage may need up to $O(N^3)$ computation time. We use W to denote the maximum number of vertices in a column in the graph after the graph is preprocessed. The number of k-cliques in the graph is at most $O(W^k)$. The computation time needed to check whether a vertex can be added into an existing clique to form a larger clique is at most $O(k)$. The computation time needed to compute the maximum weighted clique in the preprocessed graph is thus at most $O(kW^k)$. Putting the preprocessing stage and the clique enumeration stage together, the algorithm needs at most $O(N^3 + kW^k)$ computation time.

2.4. Experimental Results. We have implemented our approach with C++ programming language and used it to predict the transcription factor binding sites in the promoter regions of three genes that were selected from two organisms *Caenorhabditis elegans* and *mus musculus*. One data set is selected from the promoter region of a gene of *Caenorhabditis elegans,* and the remaining two are selected from the promoter regions of two genes of *mus musculus*. Table 2 provides a description of the testing data we have used to test the accuracy of our approach.

In order to evaluate the accuracy of our approach, we compare the binding sites predicted by our approach with the correct binding sites for all sequences in the data set. A developer score is computed to provide a measure of the accuracy. The developer score is computed based on the deviation of the predicted starting position of a binding site from its real starting position. In particular, for a binding site of length L, if the deviation is D, the developer score d of the prediction result is then computed as follows:

$$d = \max\left\{1 - \frac{D}{L}, 0\right\}. \quad (1)$$

It is not difficult to see that the developer score is a measure of the accuracy of a predicted binding site. If the predicted binding site is completely correct, the deviation D is 0, and the developer score is thus 1.0. On the other hand, if the predicted binding site is completely incorrect, it does not even intersect the real binding site, the deviation D is at least L, and the developer score is thus 0.0. In general, the developer score of a predicted binding site is a positive real number between 0.0 and 1.0. A higher value of the developer score indicates a more accurate prediction result.

We also used the Gibbs Motif Sampler [5] to predict the binding sites on the same data sets and evaluate the accuracy

Table 2: Information on the data sets we have used to test the accuracy of our approach.

Data set number	Number of sequences	Length of each sequence	Length of the binding site	Organism the data set describes
1	27	11-12	6	*Caenorhabditis elegans*
2	20	11–13	6	*mus musculus*
3	20	19–23	11	*mus musculus*

Table 3: The developer score of the predicted binding sites in data set 1 for both our approach and the Gibbs Motif Sampler. The actual binding sites are marked by nucleotides in capital letters.

Actual binding site	Binding site predicted by our approach	Binding site predicted by Gibbs Motif Sampler	Developer score of our approach	Developer score of Gibbs Motif Sampler
GAAACCctgtta	AAACCC	GAAACC	0.83	**1.00**
GAAGCCcttcaa	AAGCCC	GAAGCC	0.83	**1.00**
GAAGCCgcaaaa	GAAGCC	GAAGCC	1.00	1.00
GAAGCCcctcac	GAAGCC	GAAGCC	1.00	1.00
GAAGCCaattat	GAAGCC	GAAGCC	1.00	1.00
GAAGCCttagaa	GAAGCC	GAAGCC	1.00	1.00
GAATCCttagat	GAATCC	GAATCC	1.00	1.00
GAAACCttgcaa	GAAACC	GAAACC	1.00	1.00
GAAGCCaatcat	GAAGCC	GAAGCC	1.00	1.00
GAAACCttatga	GAAACC	GAAACC	1.00	1.00
GAAACCtttcaa	GAAACC	GAAACC	1.00	1.00
GAAACCatagac	GAAACC	GAAACC	1.00	1.00
GAAGCCttatta	GAAGCC	GAAGCC	1.00	1.00
GAAGCCcccaaa	GAAGCC	GAAGCC	1.00	1.00
GAAGCCgtagct	GAAGCC	GAAGCC	1.00	1.00
GAAGCCacaatt	GAAGCC	GAAGCC	1.00	1.00
GAAGCCgtgttt	GAAGCC	GAAGCC	1.00	1.00
GAAACCttatct	GAAACC	GAAACC	1.00	1.00
GAAGCCgtacaa	GAAGCC	GAAGCC	1.00	1.00
GAAGCCtacaaa	GAAGCC	GAAGCC	1.00	1.00
GAAACCttattt	GAAACC	GAAACC	1.00	1.00
GAAGCCgtaaaa	GAAGCC	GAAGCC	1.00	1.00
GAAGCAccttat	AAGCAC	GAAGCA	0.83	**1.00**
GAAGCCttaaaa	GAAGCC	GAAGCC	1.00	1.00
GAAGCCgtagat	GAAGCC	GAAGCC	1.00	1.00
GAAGCCactttt	GAAGCC	GAAGCC	1.00	1.00
GAATCCctacaa	GAATCC	GAATCC	1.00	1.00

of its prediction results with the developer score. We compare the accuracy of our approach with that of the Gibbs Motif Sampler. Tables 3, 4, and 5 provide the accuracy of our approach and that of the Gibbs Motif Sampler on three tested data sets in terms of the developer score.

It is evident from the testing results that our program can achieve an average developer score of 0.90 for the testing data sets. This indicates that our approach can identify the binding sites with accuracy comparable with or better than that of the Gibbs Motif Sampler for most of the tested sequences. In addition, in some cases where the Gibbs Motif Sampler has difficulty to locate the binding sites, our program can identify the binding sites with high accuracy. For example, for sequences 2, 3, and 18 in the data set for *mus musculus* that contains binding sites of length 11, the Gibbs Motif Sampler fails to identify the correct locations of binding sites while our approach identifies the binding sites for all sequences with high accuracy. The Gibbs Motif Sampler significantly outperforms our approach only in sequence 16 in the data set for *Caenorhabditis elegans*, where our approach fails to identify the correct location of binding site while the Gibbs Motif Sampler identifies its correct location without any error. It is worth mentioning that our approach achieves significantly higher prediction accuracy for all sequences in the data set for *mus musculus* that contains binding sites of length 11.

Table 4: The developer score of the predicted binding sites in data set 2 for both our approach and the Gibbs Motif Sampler. The actual binding sites are marked by nucleotides in capital letters.

Actual binding site	Binding site predicted by our approach	Binding site predicted by Gibbs Motif Sampler	Developer score of our approach	Developer score of Gibbs Motif Sampler
tgaatacCACGTG	CACGTG	CACGTG	1.00	1.00
gggatCACGTGgt	CACGTG	CACGTG	1.00	1.00
attgtgCACGTGg	CACGTG	CACGTG	1.00	1.00
CACGTGggaggtac	CACGTG	CACGTG	1.00	1.00
gggtCACGTGttc	CACGTG	CACGTG	1.00	1.00
taagCACGTGgtc	CACGTG	CACGTG	1.00	1.00
CACGTGccgcgcgc	CACGTG	CACGTG	1.00	1.00
aggtataCACGTG	TACACG	CACGTG	0.67	**1.00**
AACGTGcacatcgtcc	AACGTG	CACGTT	**1.00**	0.00
AACGTGacttcgtacc	AACGTG	CACGTT	**1.00**	0.00
CACGTGatgtcctc	CACGTG	CACGTG	1.00	1.00
CACGTGaagttgtc	CACGTG	CACGTG	1.00	1.00
AACGTGacagccctcc	AACGTG	CACGTT	1.00	1.00
agtCACGTGttcc	CACGTG	CACGTG	1.00	1.00
taaatgcCACGTG	CACGTG	CACGTG	1.00	1.00
tgaCACGTGtccg	CACGTG	CACGTG	1.00	1.00
AACGTGcgtgatgtcc	AACGTG	CACGTT	1.00	0.00
catgtCACGTGcc	CATGTC	CACGTG	0.17	**1.00**
aggaatCGCGTGc	CGCGTG	Not Found	**1.00**	0.00
agttcgCACGTGc	CGCACG	CACGTG	0.67	**1.00**

Table 5: The developer score of the predicted binding sites in data set 3 for both our approach and the Gibbs Motif Sampler. The actual binding sites are marked by nucleotides in capital letters.

Actual binding site	Binding site predicted by our approach	Binding site predicted by gibbs motif sampler	Developer score of our approach	Developer score of gibbs motif sampler
gcacATAGGTGTAAAatggccgttgg	CATAGGTGTAA	CACATAGGTGTAA	**0.91**	0.73
ctcgcacCCAGGTGTGAAgttctggt	CCCAGGTGTGA	CACCTGGGTGCGA	**0.91**	0.00
acGTAGGTGCGAAtctatcttagtgc	CGTAGGTGCGA	Not Found	**0.91**	0.00
gcgagatgtaacatGTAGGTGTGAAa	TGTAGGTGTGA	CATGTAGGTGTGA	**0.91**	0.73
ctttactcacCTAGGTGTGAATgaag	CCTAGGTGTGA	CACCTAGGTGTGA	**0.91**	0.73
gcacGTAGGTGCTACttttttgtaa	CGTAGGTGCTA	CACGTAGGTGCTA	**0.91**	0.73
acatagtgacacCTAGGTGTGAAatt	CCTAGGTGTGA	CACCTAGGTGTCA	**0.91**	0.73
cgtcacgcGTAGGTGTTACaatgtgg	CGTAGGTGTTA	CGCGTAGGTGTTA	**0.91**	0.00
gtcatGTAGGTGTGAAtatagcgccc	TGTAGGTGTGA	CATGTAGGTGTGA	**0.91**	0.73
tttgacacCTAGGTGTCATattccac	CCTAGGTGTCA	CACCTAGGTGTCA	**0.91**	0.73
tatcgcacCTAGGTGTGACaatcatc	CCTAGGTGTGA	CACCTAGGTGTGA	**0.91**	0.73
gcaaGTAGGTGTGAAatctcaacgga	AGTAGGTGTGA	CAAGTAGGTGTGA	**0.91**	0.73
acatagtgacacCTAGGTGTGAAattc	CCTAGGTGTGA	CACCTAGGTGTCA	**0.91**	0.73
gtggaatatgacacCTAGGTGTCAAa	CCTAGGTGTCA	CACCTAGGTGTCA	**0.91**	0.73
acacCTAGGTGTGAAattcagatata	CCTAGGTGTGA	CACCTAGGTGTGA	**0.91**	0.73
attagtcacacCTAGGTGTGAAgagc	CCTAGGTGTGA	CACCTAGGTGTGA	**0.91**	0.73
ccagtatcacacTTAGGTGTTACatc	CTTAGGTGTTA	CACTTAGGTGTTA	**0.91**	0.73
tctactaacagGTAGGTGTTACttgt	GGTAGGTGTTA	CAGGTAGGTGTTA	**0.91**	0.73
gcggAAAGGTGTGAAatcacaccatt	GAAAGGTGTGA	Not Found	**0.91**	0.00
gaattcacacTTAGGTGTGAAat	CTTAGGTGTGA	CACTTAGGTGTGA	**0.91**	0.73

3. Conclusions

In this paper, we developed a novel approach that can efficiently and accurately predict the transcription binding sites in the promoter regions of genes. Our approach uses a graph model to describe the subsequences in the promoter regions of homologous genes and their relationships. The problem is then reduced to a graph optimization problem. In order to efficiently compute the optimal solution of the problem, we developed a preprocessing technique that can significantly reduce the size of the graph. Our testing results on the sequence data from two organisms *Caenorhabditis elegans* and *mus musculus* showed that our approach can achieve prediction accuracy higher than or comparable with that of the Gibbs Motif Sampler.

We believe the performance of our approach can be further improved if we employ a weighted scoring scheme that can assign different relative weight values to the pair wise matching scores obtained on different positions in the subsequences. It is well known that mutation is much more likely to occur in nucleotides near the boundary of the binding sites than those near the center of the binding sites. Lower values of relative weights thus should be assigned to the matching scores obtained on nucleotides near the boundary of the binding sites. Determining the relative weights that can maximize the accuracy of prediction is an interesting problem and would be a part of our future work.

Acknowledgment

The work was supported by the Natural Science Foundation of Jiangsu Province (BK2011319).

References

[1] C. T. Harbison, D. B. Gordon, T. I. Lee et al., "Transcriptional regulatory code of a eukaryotic genome," *Nature*, vol. 431, pp. 99–104, 2004.

[2] D. J. Galas and A. Schmitz, "DNAase footprinting: a simple method for the detection of protein-DNA binding specificity," *Nucleic Acids Research*, vol. 5, no. 9, pp. 3157–3170, 1978.

[3] M. M. Garner and A. Revzin, "A gel electrophoresis method for quantifying the binding of proteins to specific DNA regions: application to components of the Escherichia coli lactose operon regulatory system," *Nucleic Acids Research*, vol. 9, no. 13, pp. 3047–3060, 1981.

[4] G. D. Stormo and G. W. Hartzell, "Identifying protein-binding sites from unaligned DNA fragments," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 86, no. 4, pp. 1183–1187, 1989.

[5] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton, "Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment," *Science*, vol. 262, no. 5131, pp. 208–214, 1993.

[6] J. S. Liu, A. F. Neuwald, and C. E. Lawrence, "Bayesian models for multiple local alignment and Gibbs sampling strategies," *Journal of American Statistics Association*, vol. 90, no. 432, pp. 1156–1170, 1995.

[7] R. Durbin et al., *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.

[8] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church, "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation," *Nature Biotechnology*, vol. 16, no. 10, pp. 939–945, 1998.

[9] S. Carey and S. Smale, "Analysis and modeling of DNA-protein interactions," in *Transcriptional Regulation in Eukaryotes: Concepts, Strategies and Techniques*, pp. 448–462, Cold Spring Harbor Laboratory Press, 2000.

[10] G. Z. Hertz and G. D. Stormo, "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences," *Bioinformatics*, vol. 15, no. 7-8, pp. 563–577, 1999.

[11] G. D. Stormo, "Computer methods for analyzing sequence recognition of nucleic acids," *Annual Review of Biophysics and Biophysical Chemistry*, vol. 17, pp. 241–263, 1988.

[12] T. L. Bailey and C. Elkan, "Unsupervised learning of motifs in biopolymers using expectation maximization," Tech. Rep. CS93-302, Department of Computer Science, University of California, San Diego, Calif, USA, 1993.

[13] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," in *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, pp. 28–36, 1994.

[14] F. F. M. Liu, J. J. P. Tsai, R. M. Chen, S. N. Chen, and S. H. Shih, "FMGA: finding motifs by genetic algorithm," in *Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering (BIBE '04)*, pp. 459–466, May 2004.

[15] D. Che, Y. Song, and K. Rasheed, "MDGA: motif discovery using a genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 447–452, Washington, DC, USA, June 2005.

[16] F. Chiaromonte, V. B. Yap, and W. Miller, "Scoring pairwise genomic sequence alignment," in *Proceedings of the Pacific Symposium on Biocomputing*, vol. 7, pp. 115–126, 2002.

Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2012, Article ID 892098, 5 pages
doi:10.1155/2012/892098

Research Article

A Dynamic Data-Driven Framework for Biological Data Using 2D Barcodes

Hui Li and Chunmei Liu

Department of Systems and Computer Science, Howard University, Washington, DC 20059, USA

Correspondence should be addressed to Hui Li, hli3302@gmail.com

Received 9 September 2012; Accepted 4 October 2012

Academic Editor: Xumin Liu

Biology data is increasing exponentially from biological laboratories. It is a complicated problem for further processing the data. Processing computational data and data from biological laboratories manually may lead to potential errors in further analysis. In this paper, we proposed an efficient data-driven framework to inspect laboratory equipment and reduce impending failures. Our method takes advantage of the 2D barcode technology which can be installed on the specimen as a trigger for the data-driven system. For this end, we proposed a series of algorithms to speed up the data processing. The results show that the proposed system increases the system's scalability and flexibility. Also, it demonstrates the ability of linking a physical object with digital information to reduce the manual work related to experimental specimen. The characteristics such as high capacity of storage and data management of the 2D barcode technology provide a solution to collect experimental laboratory data in a quick and accurate fashion.

1. Introduction

With the development of various biological techniques, the study of complex biological mechanism on the molecular level is close to disclose the reality of life. Biological experiment always provides the most powerful first-hand evidence of the changes or discovery of biological systems. Currently, conforming experimental cohort, sampling, tagging, freezing specimens, selecting experimental instrument, and generating experimental data are time consuming, error prone, and laboratory extensive for biologists. Particularly, the changes of the previous experimental schema will cause lots of unexpected errors or troubles. Data grows exponentially every day; a system solution is far from the utility. It is necessary and urgent to propose an efficient computational approach to systematically manage and simplify the whole process to improve biology data management and to eliminate potential errors as well as save time [1–3]. Efficiently managing experimental data is always the first demand for biologists. For instance, in most cases, biologists plan the experimental schema and detailed steps by hand before they start to do the experiments. Knowledge about the results to be obtained experimentally is also queried manually from online databases or websites. Once they finish the experiments, they have to tag manually the data generated from the experiments for further analyzing the data. At the same time, paper work to describe the details of the specimen and experimental schema has to be prepared and input manually into a computer, which may lead to some errors. In particular, the different biological specimen sampling methods and standards used in various experimental laboratories result in the difficulty of reproducibility of the experiments as well as exchanging data among different groups. With the availability of the computational recognition and image processing technology, the 2-dimensional (2D) barcode technologies provide an easy solution for the experiment management and data analysis, track, and management. 2D barcode is a type of automatic recognition method by attaching encoded information on the specimen and dataset [4]. 2D barcode is one of the popular techniques and widely applied in multiple fields such as products anti-counterfeit, map guidance, health system, and websites [5–7]. Although the method has been applied successfully in these fields, exchanging data among different groups is impossible for

sharing data among different laboratories. Obviously, the inconsistence of the protocols between different phrases of the experiments increases the difficulty of the 2D barcode recognition system. Therefore, the implementation of this technology firstly needs the consistence and standard of protocols for data collection, biological specimen tagging, slides, and specimen containers.

Motivated by these challenges, we apply the 2D barcode technique and extend its ability to deal with experimental-related information to draft experimental schema, tag data, and store experimental protocol. We thus propose a data-driven laboratory information system which inspects laboratory equipment and eliminate impending failures. We put 2D barcodes on each specimen and input all the information in it which is to be recognized easily by any electronic scanner such as a mobile phone or a computer or any equipment with a camera. Also some image information can be stored in the 2D barcodes. The experiments show the proposed data-driven laboratory information management system based on the 2D barcode technique which improves the efficiency and reliability of the laboratory management system without the involvement of biologists.

The paper is organized as follows. The overview section introduces the proposed method where the 2D barcode technique is the key of this section. In the method section, we introduce a data-driven framework which can reduce the computational process and minimize total delay time. The results section shows the capability of the proposed method. In conclusion and algorithms section, we summarize the method.

2. Dynamic Data-Driven Framework for a Biological System

The concept of dynamic data-driven framework dynamically creates computational pipelines and is trigged by the biological data. Since the large volume of experiment data produced from the laboratories need to be processed for biologists, most of biology experimental dataset-related information is complex and heterogeneous. In order to integrate heterogeneous data, we use XML as the 2D encode barcode format. All the information is structured by using the XML format and we build an xml parsing model to parse and decode the 2D barcode images. 2D barcodes can be read by optical scanners with special software. In order to analyze the raw data, there are a computational pipeline and a database to analyze the raw data and store them, respectively. For our data-driven framework, each pipeline is created and trigged when it receives 2D barcode information.

2.1. Framework Model. The framework used in this paper is shown in Figure 1. The XML module transforms the input file into a QR support encoding format, and then the specimen is tagged by the 2D barcodes with the QR codes. The information of the 2D barcodes is saved into DBXML [8] at the same time. The event trigger is also written in the 2D barcodes. When the users scan the 2D barcodes and read the trigger event, the XML parser will parse the xml file to start the pipeline.

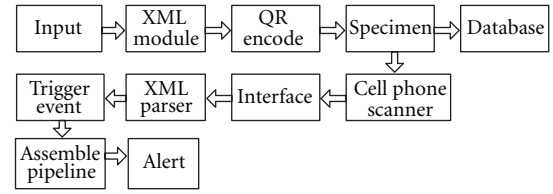


Figure 1: The model of the service-oriented dynamic data-driven framework.

2.2. Application of the System to Biological Systems. In this section, we describe the data-driven framework on biological systems. The framework includes four main components. Figure 1 shows the four components used in the framework.

(1) *The first part is the identification* of specimen using the 2D barcode technique. It collects all the detailed information of the specimen by a scanning device. The user will then get a list of all the barcodes and it also facilitates the interactions with the 2D barcode reader such as a smart phone.

(2) *The second part is the pipeline module*; the data scanned by the 2D barcode is a trigger for driving the pipeline which directly obtains the required data and information.

(3) *The third part is the detection module*, which will monitor and detect the congestion node of the pipeline and then alert the user the next data congestion point. If it finds any data need further experimental steps, it will then notify the experiment staff to finish the experiment.

(4) *Using a heuristic algorithm t*o find paths of the pipeline by calculating the weights of nodes in the pipeline and trigger the most efficient pipeline.

3. Algorithms

In this paper, there are two types of dynamic biology data (raw and refer data) as mentioned in the above section. We first embed the information of a specimen item in a barcode and the information like the expiration date can be easily seen. In order to avoid the congestion of the pipeline and find the path out of the pipeline, we proposed a heuristic algorithm and assembling pipeline strategy.

When a biologist parses the XML and checks the data completeness, he first checks the entire list of the specimen scanned and finds the node of the pipeline can be triggered and dynamically assembles the pipeline. The heuristic algorithm ensures the above steps to be included sequentially. The detailed description of the algorithm is as follows.

Step 1 (constructing the data-driven mechanism). Providing the standard formatted specimen information to generate a 2D barcode image as shown in Figure 2, 2D barcodes save the information of the specimen to be identified. In this paper, we use the XML format to encode the 2D barcodes for experimental specimen information. We define the nodes

```
RawDataSchema
<?xml version="1.0" encoding="ISO-8859-1"?>
<specimen id="89923">
    <specimenname>Mass spectra</specimenname>
  <experimentperson>John Smith</experimentperson>
  <externaldatabase>false</externaldatabase>
<source>
    <address> new street</address>
    <city>marryland</city>
    <country>USA</country>
</source>
<item>
  <seq1>AHR</seq1>
  <seq1>AHR</seq1>
</item>
ReferDataSchema raw data stored in the DBXML, can be retrieved by KeyID.
<?xml version="1.0" encoding="ISO-8859-1"?>
<specimen id="89723">
    <specimenname>Mass spectra </specimenname>
  <experimentperson>John Smith</experimentperson>
  <externaldatabase>false</externaldatabase>
<source>
  <address>new street</address>
  <city>marryland</city>
  <country>USA</country>
</source>
<item>
      <location1>
    <KeyID>
       012222
    </KeyID>
     </location1>
   </item>
```

Algorithm 1: Experiment data format.

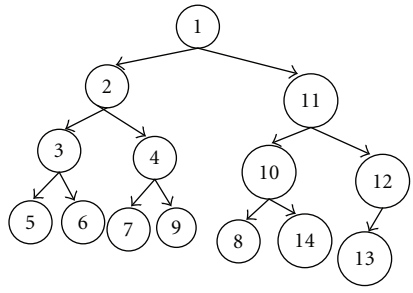


Figure 2: The process of the path search.

of the pipeline and the triggering event which are embedded in the 2D barcodes. The nodes of the pipeline include *R* functions, raw data uploading, data preprocessing, and Matlab functions. The pipeline starts from the event processing. The operations include data verification, paths of pipeline checking and pipeline assembly. Since 2D barcodes contain considerably less than 2000 characters, we divide the 2D barcode data into two categories: (1) *raw data* which means all the data from biological experiments can be stored in the barcode and (2) *refer data* which indicates that the data of the experiment exceeds the limit of the 2D barcode, and then the 2D barcode represents the description of the raw data and the location of the data for external database. In this paper, we use a XML file for encoding information inside a 2D barcode. We will describe the format that is generated for specific purposes shown in experiment data format (Algorithm 1).

Step 2 (pipeline construction). The second step is to define the node of the pipeline in the system and determine the weight value for each node according to the data and the final goal. The pipeline is built based on the data and bioinformatics program analysis and the pipeline consists of in-house programs, raw data from the experiment, intermediate results, and tools, which are illustrated in description of algorithm (Algorithm 2).

We assume that the pipeline congestion is caused by lacking of data or waiting for other processing and each node relies on the output of the previous node in the pipeline tree.

```
Pipeline tree algorithm
Arraylist pipeline
Arraylist lackdata
While(queue!=empty)
{
        i=getcurrentnode();
        forall edge(i,j)
        {
            if novisited(j)
            {
                V_j=E time(i,j)
                }
                pipeline =Addpieline()
        }
}
Alert(lackdata)
Return pipeline.
```

Algorithm 2: Description of algorithm.

Each node has four possible directions (left, right, up, and down). Consequently, the raw data of the node can be moved along to a goal node. The weighted value of the node is defined according to the following:

$$E\,\text{time}(i,j) = \sum_{1}^{m} D_t(k) + Dis(i), \tag{1}$$

where i and j are the sequence number representations of the node in the pipeline tree, $D_t(k)$ is the waiting time for the data that the node requires, and m is the number of required data between the node i and node j.

Step 3 (assemble a pipeline). The aim of this step is to obtain the effective pipeline tree for the raw data. In order to minimize total delay time for the computational process of the pipeline, we use the Dijkstra algorithm [9] to calculate the accumulative weights and find an efficient pipeline out of all possible nodes. In the Dijkstra process, we select the highest priority node and move it from the raw data node to the goal node and obtain a shortest path by calculating the value in the pipeline tree. The algorithm will output a pipeline tree and alert users about the lack of data.

4. Experiments

In order to test the performance of our framework, we generate simulation data for the pipeline, and then used our method to run Table 1 shows the simulation results from the raw data. In each experiment, Dijkstra approaches have been executed for searching the most efficient pipeline tree based on the trigger event. If the results of two prediction strategies have the same accuracy and efficiency, the difference between these average travelling times should be close to zero.

Table 1: The simulation results of the data-driven framework.

Number of nodes of a pipeline	Data-driven framework	No data-driven framework
40	Dijkstra 15 hours	No algorithm 21 hours
	Improvement (21−15) hours/21 = 28.57%	

5. Conclusion

In this paper, we propose a data-driven framework to improve the management efficiency of the biology laboratory system. It encodes the detailed information of a specimen in the 2D barcodes and embeds the trigger event for a pipeline. The experimental results show that the proposed method improves system efficiency by processing a fraction of a large volume of data. It not only promotes the efficiency of the biological system, but also reduces the error of the system.

Acknowledgments

This work was supported by NSF CAREER (CCF-0845888) and by the Center for Science of Information (CsoI), NSF Science and Technology Center, under Grant Agreement CCF-0939370.

References

[1] S. Tanner, S. H. Payne, S. Dasari et al., "Accurate annotation of peptide modifications through unrestrictive database search," *Journal of Proteome Research*, vol. 7, no. 1, pp. 170–181, 2008.

[2] E. W. Deutsch, H. Lam, and R. Aebersold, "Data analysis and bioinformatics tools for tandem mass spectrometry in proteomics," *Physiological Genomics*, vol. 33, no. 1, pp. 18–25, 2008.

[3] A. Andreeva, D. Howorth, J. M. Chandonia et al., "Data growth and its impact on the SCOP database: new developments," *Nucleic Acids Research*, vol. 36, no. 1, pp. D419–D425, 2008.

[4] J. Z. Gao, L. Prakash, and R. Jagatesan, "Understanding 2D-BarCode technology and applications in M-commerce—design and implementation of A 2D barcode processing solution," in *Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC '07)*, pp. 49–56, July 2007.

[5] M. López-Nores, J. J. Pazos-Arias, J. García-Duque, and Y. Blanco-Fernández, "Monitoring medicine intake in the networked home: the iCabiNET solution," in *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare*, pp. 116–117, PervasiveHealth, February 2008.

[6] T. Y. Liu, T. H. Tan, and Y. L. Chu, "2D barcode and augmented reality supported english learning system," in *Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS '07)*, pp. 5–10, July 2007.

[7] S. Lisa and G. Piersantelli, "Use of 2D barcode to access multimedia content and the web from a mobile handset," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 5594–5596, December 2008.

[8] Oracle Berkerly DBXML database, http://www.oracle.com/technetwork/products/berkeleydb/index-083851.html.

[9] R. Estrada, C. Tomasi, M. T. Cabrera, D. K. Wallace, S. F. Freedman, and S. Farsiu, "Exploratory Dijkstra forest based automatic vessel segmentation: applications in video indirect ophthalmoscopy (VIO)," *Biomedical Optics Express*, vol. 3, pp. 327–339, 2012.

Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2012, Article ID 135780, 4 pages
doi:10.1155/2012/135780

Research Article

Biomarker Identification Using Text Mining

Hui Li and Chunmei Liu

Department of Systems and Computer Science, Howard University, Washington, DC 20059, USA

Correspondence should be addressed to Hui Li, hli3302@gmail.com

Received 9 September 2012; Accepted 4 October 2012

Academic Editor: Xumin Liu

Identifying molecular biomarkers has become one of the important tasks for scientists to assess the different phenotypic states of cells or organisms correlated to the genotypes of diseases from large-scale biological data. In this paper, we proposed a text-mining-based method to discover biomarkers from PubMed. First, we construct a database based on a dictionary, and then we used a finite state machine to identify the biomarkers. Our method of text mining provides a highly reliable approach to discover the biomarkers in the PubMed database.

1. Introduction

Identifying molecular biomarkers has become an essential task for bioinformatics scientists to assess the different phenotypic states of cells or organisms correlated to the genotypes of diseases from large-scale biological data [1]. The text mining technique has become a critical technique for designing future predictive and personalized medicine. At the same time, the PubMed database which comprises more than 21 million citations for biomedical literature offers an enriched source for us to explore the biomarkers across human disease and to mine the biomarkers related to diseases. Therefore, integrating automatic literature searches, and text mining is a fast emerging research area in epigenetics, DNA methylation, and more specifically biomarker discovery studies. For almost every cancer type, new publications that discover biomarker candidates are updated frequently, especially with advanced high-throughput methodologies. Efficient text mining tools and algorithm development are extremely needed.

Many text mining technologies that are proposed by different groups, such as machine learning technologies including support vector machine [2], decision tree [3], Bayes classifier [4], and random forest [5], are used for text mining. Also the natural language processing technique is used to determine the structures and linguistic components of sentences and then parses the sentences in a bag of words, together with a statistic approach to get the matched results from the text databases. OMIM database [6] is one of the important databases for biomarker-related disease research. The MeSH Browser [7] is used to map disease associations to MeSH IDs.

In this paper, we use a state machine to simulate the transforms of the biomarkers from individual entities to associated diseases and pathways as well as networks. Several abstracted templates are summarized from known expert experience and knowledge. The biomarkers are ranked based on the importance to the diseases and the citations of the literature from PubMed. Based on this template, every mined biomarker-related pathways, networks, and disease will be collected and matched with the templates.

2. Method

All the biomarkers mentioned in this paper are mined from the PubMed database. For each biomarker candidate, we use a finite state machine (FSM) [8] to identify biomarker, pathways, and associated diseases. Only the candidates which are accepted by FSM are viewed as biomarkers. The association between the biomarkers and the diseases can be output to refine the biomarkers.

As shown in Figure 1. The first step is to create a biomarker dictionary, the second step is construct a DBXML [9] database, and the third step is using the finite state machine to conform the disease-related biomarkers. We first create our DBXML database from the PubMed database. The Lucence technique is used to split the document into a bag

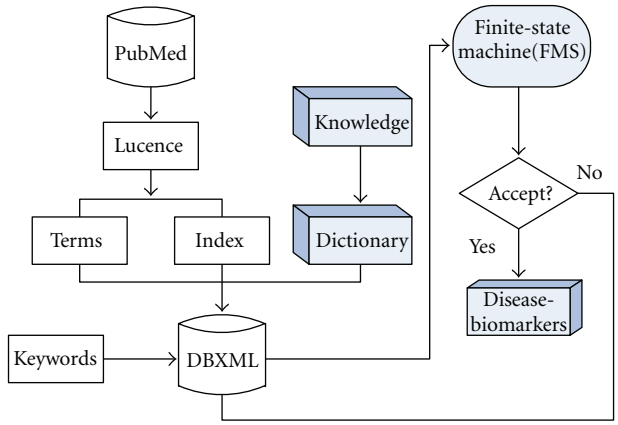


Figure 1: The flow chart of the biomarker discovery.

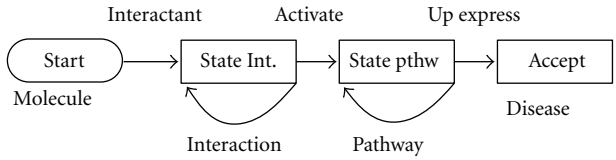


Figure 2: The identification of the biomarker using the finite state machine.

Table 1: The dictionary of the biomarkers.

Gene	Protein	Pathway	Disease
P53	P53	Ras	Diabetes
APC	APC	Wnt	Breast cancer
MDM2	Pten	Death receptor pathway	Liver cancer
Ras	HCC	Ether lipid metabolism	Huntington
Axin-1	HPR	Thiamine metabolism	Liver cirrhosis
	LCE2B	Porphyrin and chlorophyll	Prostate cancer
	AXIN1	Metabolism	Leukemia
	SLC22A1		

of words and extract terms such as gene names, interaction relationships, pathways, and network names. In the mean time, based on our domain knowledge, we construct the dictionary for further analysis. Based on the Lucence parsed terms and dictionary, the DBXML database is created for biomarker extraction. To retrieve the keyword from the DBXML database, exact matching, fuzzy matching, and list matching methods are used to match the terms saved in DBXML. If the end state of the FMS is in an acceptance state, the keywords-related genes, proteins, or small molecules are marked as biomarkers.

2.1. Construct a Database for Gene/Protein and Disease. We first construct a database which includes categories according to their names, diseases, interactions, pathways, and network information. Then, we collect a list of diseases, gene/protein and so on, and then put them into the dictionary. The structure of the dictionary is shown as Table 1.

We use a dynamic method to collect the full-text document, and then the Lucence is applied to split the word. For Lucence, we need to delete the old document and create new Lucence document index. The Lucence document contains three paths, content and the index of the document, the terms, and the modified date.

Each word is separated by a series of phrases, and we use the dictionary to parse the full-text and then divide them into several primary categories: molecule names, interaction keywords, and verbs. After we extract the keywords, we construct the segment of the xml document for those keywords. The protein name is an entity, and the interaction represented the relation of the entity which is used to extract the relationships between diseases, genes, mutations, and proteins. We give an example of an xml segment extracted from PubMed as follows:

```
<protein id=010>
    <name>P53    </name>
    <interact>MDM2</interact>
</protein>
```

If the words cannot match the dictionary, it will be ignored. Some keywords can be removed from the database as they are not suitable for our definition. Additional tags can also be added by the users. Table 1 shows the dictionary of the biomarkers.

Our database does not contain interaction pairs and pathways. We will dynamically parse online databases for the protein/gene names and build the interaction network.

2.2. Using the FSM to Identify Biomarkers. We used the finite state machine (FSM) to identify the biomarkers in our database. The FSM is a state machine which has a start node, accepting node, input entities, and relations. The roles contain the information of each entity such as genes, proteins, and small molecules.

In this paper, the FSM for identifying biomarkers is regarded as a template which serves to match corresponding biomarkers as shown in Figure 2. In addition, the template can be modified by users. Our methods include exact

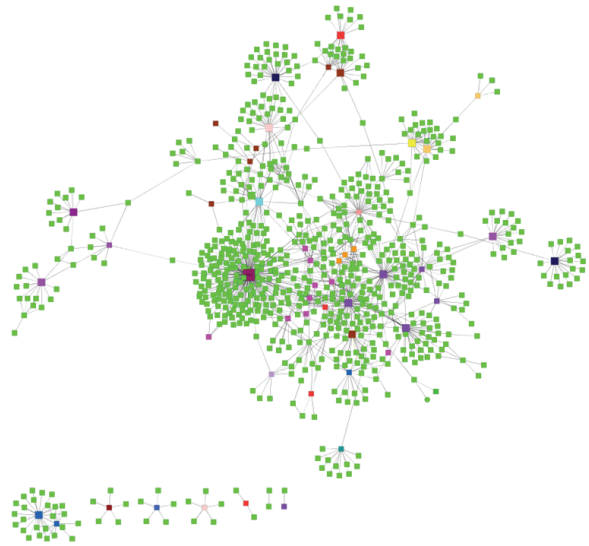


Figure 3: A diseases-related gene associated network. Green nodes are genes, and the nodes in other colors are diseases.

Table 2: The list of biomarker-disease associations mined from PubMed.

EntrezID	Gene name	Symbol
11914	ALPHA 1,4-GALACTOSYLTRANSFERASE	A4GALT
3558	ACETOACETYL-COA SYNTHETASE	AACS
5758	ABHYDROLASE DOMAIN CONTAINING 1	ABHD1
18925	ACYL-COA THIOESTERASE 12	ACOT12
18925	ACYL-COA THIOESTERASE 12	ACOT12
17809	ACYL-COA THIOESTERASE 2	ACOT2
17766	ACYL-COA THIOESTERASE 4	ACOT4
15426	ACYL-COA SYNTHETASE BUBBLEGUM FAMILY MEMBER 1	ACSBG1
11191	ACYL-COA SYNTHETASE BUBBLEGUM FAMILY MEMBER 2	ACSBG2

matching, fuzzy matching and list matching. For disease, we use the exact match method, for all molecules, we use fuzzy matching, and for interaction, we use list-member matching.

For the list member of interactions, the list members are defined as $\mathrm{ILIST}(P_a) := (P_1, P_2, P_3, \ldots P_n)$, where P_a interacts with $P_1, P_2, \ldots P_n$, which dynamically parse online databases. We construct the protein-protein interaction network around P_a in the FSM. We also obtain the pathway from the KEGG database.

The FSM includes <left-context FSM>, <entity FSM>, and ⟨right-context-FSM⟩. The roles of the entities are determined by the context of the left and right neighbors of the entities.

For example, for the entity P53 which is a protein, we determine the role of the entity as follows:

If < right-context >

=<(“express′′“present′′)(“in′′“at pathway′′)>

Then entity role = in the pathway

The output of the FSM is the track nodes between <Disease>⟺<Potential Biomarker> which include paper name and author name. The FSM is shown in Figure 2.

3. Experimental Results

Based on our framework, a query on liver cancer and the candidate biomarkers are report as Table 2.

In a query process, we dynamically parse the identified genes/proteins and construct the interact network. We then use Cytoscape software [10] to display the interaction network shown as Figure 3.

4. Conclusions

The proposed method is based on text mining technique from the PubMed database, combined with the full text search-engine technology (Lucence), a complex network of biological and signaling pathways. First, we construct a database based on a dictionary; second, we use a FSM

to identify the biomarkers; finally, we output the disease-associated biomarkers. This research offers a comprehensive text mining to discover biomarkers.

Acknowledgments

This work was supported by NSF CAREER (CCF-0845888) (H. Li and C. Liu) and by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under Grant Agreement CCF-0939370.

References

[1] Y. Chervona and M. Costa, "Histone modifications and cancer: biomarkers of prognosis?" *American Journal of Cancer Research*, vol. 2, no. 5, pp. 589–597, 2012.

[2] Z. Qi, Y. Tian, and Y. Shi, "Laplacian twin support vector machine for semi-supervised classification," *Neural Network*, vol. 35, pp. 46–53, 2012.

[3] E. Taniguchi, T. Kawaguchi, M. Sakata, M. Itou, T. Oriishi, and M. Sata, "Lipid profile is associated with the incidence of cognitive dysfunction in viral cirrhotic patients: a data-mining analysis," *Hepatology Research*. In press.

[4] H. Zhang, G. Liu, T. W. S. Chow, and W. Liu, "Textual and visual content-based anti-phishing: a Bayesian approach," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1532–1546, 2012.

[5] W. G. Touw, J. R. Bayjanov, L. Overmars et al., "Data mining in the Life Sciences with Random Forest: a walk in the park or lost in the jungle?" *Briefings in Bioinformatics*. In press.

[6] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, "Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders," *Nucleic Acids Research*, vol. 33, pp. D514–D517, 2005.

[7] M. Crespo Azcarate, J. Mata Vazquez, and M. Mana Lopez, "Improving image retrieval effectiveness via query expansion using MeSH hierarchical structure," *Journal of the American Medical Informatics Association*. In press.

[8] M. Garcia-Remesal, V. Maojo, and J. Crespo, "A knowledge engineering approach to recognizing and extracting sequences of nucleic acids from scientific literature," in *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1081–1084, 2010.

[9] Oracle: Oracle Berkeley DB XML, 2012.

[10] P. Shannon, A. Markiel, O. Ozier et al., "Cytoscape: a software Environment for integrated models of biomolecular interaction networks," *Genome Research*, vol. 13, no. 11, pp. 2498–2504, 2003.

Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2012, Article ID 696190, 6 pages
doi:10.1155/2012/696190

Research Article

Identification of Novel Type III Effectors Using Latent Dirichlet Allocation

Yang Yang

Department of Computer Science and Engineering, Information Engineering College, Shanghai Maritime University, 1550 Haigang Avenue, Shanghai 201306, China

Correspondence should be addressed to Yang Yang, yangy09@gmail.com

Received 14 May 2012; Revised 7 August 2012; Accepted 12 August 2012

Academic Editor: Chunmei Liu

Among the six secretion systems identified in Gram-negative bacteria, the type III secretion system (T3SS) plays important roles in the disease development of pathogens. T3SS has attracted a great deal of research interests. However, the secretion mechanism has not been fully understood yet. Especially, the identification of effectors (secreted proteins) is an important and challenging task. This paper adopts machine learning methods to identify type III secreted effectors (T3SEs). We extract features from amino acid sequences and conduct feature reduction based on latent semantic information by using latent Dirichlet allocation model. The experimental results on *Pseudomonas syringae* data set demonstrate the good performance of the new methods.

1. Introduction

Secretion is an essential mechanism for bacterial adaptation and survival in their surrounding environment. The secretion process transports effector molecules from the interior of a bacterial cell to its exterior. Up to now, researchers have discovered six types of secretion systems. The type III secretion system is one of the most complex ones, which allows bacteria to deliver virulence effectors across eukaryotic cellular membranes [1].

In recent years, significant progress has been made in our understanding of the structural components of T3SS, including a needle-like component and bases embedded in the inner and outer bacterial membranes [2]. However, the details of the secretion mechanism and defined signals remain unknown. Identification of the effectors secreted by the T3SS (called type III secreted effectors, T3SEs) is very important to the T3SS study. They are believed to have some unique characteristics that can be recognized by the secretion system and be delivered into host cells. These characteristics are hints to uncover the mechanism of T3SS and understand the role that each component plays in the secretion process.

The amino acid sequences of T3SEs have great sequence diversity through fast evolution, and many T3SEs have very few homologous proteins in the public databases. Therefore, it is notoriously challenging to recognize T3SEs. The plant pathogen *Pseudomonas syringae* has been a model for the research of type III effectors. Thus far, only several hundreds of T3SEs have been identified and confirmed from all the bacterial species, and a large portion of them are from *P. syringae* strains. It indicates that a vast majority of T3SEs remain unknown.

This study aims to develop a computational prediction system, which can help the biologists to obtain the effector candidates for wet-bench experimental confirmation. Generally, the computational tools for predicting T3SEs can be divided into two types: sequence-based and domain knowledge-based.

The sequence-based methods usually attempt to extract discriminant subsequence features from amino acid sequences or nucleotide sequences and perform prediction based on these features. The features extracted from amino acid sequences include amino acid composition, K-mer frequencies [3, 4], and position-specific features [5]. As for the nucleotide sequences, genes encoding the T3SS apparatus and T3SEs usually have a conserved regulatory motif in their promoters [6]. Another sequence-based method, homology search using known effectors [3], is also often used, but it

cannot identify novel effectors. The domain-knowledge-based methods include identifying genes in vicinity to chaperone homologues [7], predicting instability of N-terminus and nonoptimal codon usage [8], and using protein secondary structure and solvent accessibility information [9]. The domain knowledge is not as available as sequence data and usually obtained by computational approaches, which lowers the prediction accuracy.

This paper adopts machine learning methods to predict type III secreted effectors (T3SEs). The features are extracted from amino acid sequences. Researchers have detected amino acid composition biases in T3SEs, especially in the N-termini. For example, Guttman et al. [10] reported that the first 50 amino acids of *P. syringae* effectors have a high proportion of Ser and a low proportion of Asp residues. It should be noted that these observations only reveal some statistical biases instead of specific signals/features. Moreover, many effectors do not fulfill these requirements. In this paper, we regard the protein sequences as a kind of biological language and the K-mers as words. The word frequencies compose the feature vectors. In order to condense the feature space and improve the prediction accuracy, we propose two feature reduction methods. Both of them utilize latent semantic information in the latent Dirichlet allocation model [11].

We have examined the prediction accuracies of these two methods and compared them with four other methods, including dimer frequency, trimer frequency, and feature selection using frequency as well as $tf\text{-}idf$ value. The methods were tested on the *Pseudomonas syringae* data set through fivefold cross-validation. The experimental results demonstrate the effectiveness of the proposed methods.

2. Methods

Protein sequences are consecutive amino acid residues, which can be regarded as text strings with an alphabet $\mathcal{A}$ of size $|\mathcal{A}| = 20$. The amino acid composition and K-mer (subsequence with length of K) frequency can be used as features for the protein sequence classification. The amino acid composition does not consider the order of amino acids, while K-mers retain some sequence order information, thus the latter method is usually adopted. However, the dimensionality of K-mer feature space grows exponentially as K increases. The prediction based on the full K-mer feature space without any dimension reduction would be computationally intractable. In fact, lots of K-mers are irrelevant to the prediction. For example, the K-mers appear only once or very few times.

In this paper, we propose two feature reduction methods based on latent Dirichlet allocation (LDA) model [11]. These two methods utilize the latent semantic information in different ways. One is to convert the original K-mer space to topic space, and the other is to use topic information to select informative K-mers for prediction. These two methods are introduced in Sections 2.2 and 2.3, respectively.

2.1. Latent Dirichlet Allocation. Latent Dirichlet allocation (LDA), the most common topic model currently in use, has been widely applied in natural language processing, image classification, social network analysis, and so forth [12, 13]. In LDA model, each document can be viewed as a mixture of various topics and that each word's creation is attributable to one of the document's topics.

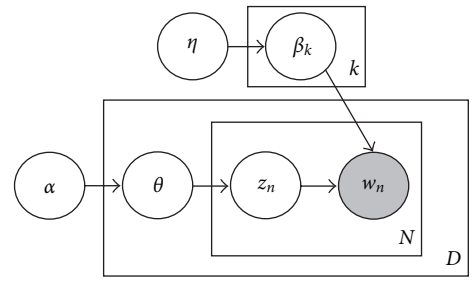


Figure 1: An LDA model.

Figure 1 shows a graphical model representation of LDA. (Here we consider the smoothed LDA.) The square frames represent replicates. There are D documents in the corpus, N words, and K topics. In this LDA model, a document is generated as the following steps.

Draw θ from the Dirichlet prior:

$$\theta \sim \mathrm{Dir}(\alpha). \tag{1}$$

For each word w_n, pick a topic z_n from multinomial(θ), and then pick w_n from $p(w_n \mid z_n, \beta)$, which is a multinomial probability conditioned on the topic z_n:

$$\begin{aligned} z_n &\sim \mathrm{Mult}(\theta) \\ w_n &\sim p(w_n \mid z_n, \beta). \end{aligned} \tag{2}$$

The likelihood of generating a corpus $\mathcal{D}$ is defined in the following equation:

$$p(\mathcal{D} \mid \alpha, \eta) = \iint \prod_{k=1}^{K} p(\beta_k \mid \eta) \prod_{d=1}^{D} p(\theta_d \mid \alpha) \times \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \right) d\theta \, d\beta. \tag{3}$$

In this model, only w_n is fully observable. Inference of the hidden variables often adopts Gibbs sampling [14] or variational algorithms [15]. Since LDA is a generative model, with limited discriminative ability in classification tasks, we only use it for feature creation.

2.2. Prediction of T3SEs in the Topic Space. In LDA model, each document is represented by a posterior Dirichlet over the topics. This is a much lower dimensional representation compared with using word frequency. Therefore, in this method, we create feature vectors by using the topic representation.

We regard protein sequences as text, and the K-mers are words. We would like to use LDA model to catch the latent

topic information. Since the LDA model cannot be used directly on the protein sequences, we need first convert the protein sequences to a kind of biological language, whose words are the K-mers. Similar to Chinese sentences, we segment the amino acid sequences to nonoverlapping K-mers without spaces between words. After that, the LDA model can be applied on the sequences.

All the natural languages have predefined dictionaries. However, protein sequences are written in an unknown language to us at the present state, whose words are not delineated. Any combination of letters with arbitrary length may be a word. So we first need to build a dictionary, which is the basis of segmentation. Therefore, the whole process of this method consists of three steps: (1) construct a dictionary, that is, word set; (2) segment the protein sequences by matching the words in the dictionary, that is, K-mers; (3) run LDA model on the segmented sequences and create feature vectors.

We have tried two measures to determine the words to be included in the dictionary. One is word frequency, and the other is tf-idf value. They are defined in the following.

2.2.1. Frequency. In natural language, words are generally the combinations of characters that frequently appear in the text. According to this observation, the amino acid substrings with high frequencies can be regarded as words, which should be segmented out and used as features. The unusual strings are noninformative for classification and have little influence on global performance. We record the appearance time for each K-mer in the training sequence set and preserve a predefined proportion of the most frequent K-mers.

2.2.2. tf-idf Value. Considering that the frequency measure is apt to select the overrepresented words in the text, which may have little discriminant ability, we also use the tf-idf value. According to its definition in text categorization, tf-idf is calculated for a term in a single document. The value is in proportion to the number of occurrences of the term in the document, that is, the tf (term frequency) part; and in inverse proportion to the number of documents in the training set for which the term occurs at least once, that is, the idf (inverse document frequency) part.

Here we redefine it as the following equation. Let $w_{t,s}$ be the tf-idf value for a K-mer t in sequence s, $f_{t,s}$ be the frequency of K-mer t in sequence s, N be the size of the training set, and n_t be the number of sequences in which t appears:

$$w_{t,s} = f_{t,s} \times \log \frac{N}{n_t}. \tag{4}$$

To avoid encountering unknown words, all 20 amino acids are included in the dictionary.

In the second step, we used the segmentation method proposed in [16]. This segmentation method has two criteria in searching the best way of segmentation. One is that the number of segments is the smallest. The other is that the product of the weights of the words segmented out is the biggest.

If the frequency measure is used in dictionary construction, the weight of word t is defined by frequency as follows:

$$w_t = \sum_{s=1}^{N} f_{t,s}. \tag{5}$$

Or else, if the tf-idf measure is used, the weight of word is defined as the maximum value of $w_{t,s}$, which is the tf-idf value for a K-mer t in sequence s:

$$w_t = \max_{s \in \mathcal{T}} w_{t,s}, \tag{6}$$

where $\mathcal{T}$ denotes the whole data set.

After segmentation, we run LDA model on the sequences. Then we obtain a sparse $D \times T$ matrix A, where D is the number of sequences and T is the number of topics. $A(d, j)$ contains the number of times a word token in document d has been assigned to topic j. The row vectors are the feature vectors used in the classification. Here, we classify the protein sequences in the topic space instead of word space. Thus the dimensionality of the feature set can be greatly reduced because the number of topics are much less than that of words.

2.3. Prediction of T3SEs in the Reduced Word Space. In this method, the feature representation is totally different from the first method. Here we still use K-mer frequencies as features. Instead of using all the K-mers in the dictionary, we select informative ones according to the topic information.

The feature reduction process also consists of three steps. The first two steps are the same as in Section 2.2, while the third step needs certain strategies for selecting words.

Actually, the dictionary construction can be regarded as the initial screening procedure for word selection. The appearance times of the words in the dictionary can be recorded and compose the feature set. In the experiments, we examined the prediction accuracies of these two kinds of feature sets using frequency and tf-idf for word selection, respectively, and find that the frequency is better than tf-idf in this study (see the results shown in Table 2). Thus we conduct the third step based on the dictionary constructed by the criterion of word frequency.

Here we perform a further selection using topic information. We examine the number of times that words are assigned to topics and set a threshold m. If a word is not assigned to any topic at least m times, this word is discarded. In this way, we could remove the words which are either unusual words or not specific to any topic.

2.4. Complexity Analysis. The computational time is mainly spent on sequence segmentation and LDA model. The segmentation algorithm [16] regards each amino acid as a check point. At each point, the algorithm conducts pruning by keeping only the optimal segmentation which has the least number of segments up to the current point, and search words by matching the subsequences next to the point with the words in the dictionary. Suppose that the dictionary size is S, the number of protein sequences in the data set is D,

Table 1: Data distribution.

Dataset	Number
Positive set	108
Negative set	760
Total	868

the average sequence length is L, and the maximum length of words is M, matching a word in the ordered dictionary has a cost of $O(\log 2S)$ by binary search. Thus, the computational complexity of the segmentation method is $O(DLM \log 2S)$ ($M = 3$ in the experiments). As for the LDA model, suppose there are K topics, the complexity is $O(DKL)$ using Gibbs sampling technique for parameter estimation and inference. And in the second feature selection method, the complexity of selecting words is $O(KS)$.

3. Results and Discussion

3.1. Data Set. Since *Pseudomonas syringae* has been used as a model organism in the study of T3SEs, it has the most effectors that have been confirmed. Therefore, we collected data from this species. To our knowledge, there is a total of 283 effectors, have been confirmed, from *P. syringae* pv. tomato strain DC3000, *P. syringae* pv. syringae strain B728a, and *P. syringae* pv. phaseolicola strain 1448A. However, a large portion of them are homologs, that is, the sequence similarity is very high. This is because the homology-based search is still the major means to discover novel effectors. Considering that the redundancy of the data set would result in overestimation on the accuracy of the classifier, we eliminated the samples with sequence similarity over 60%. By removing the redundant sequences, we get a positive set of 108 samples.

The negative data set was extracted from the genome of *P. syringae* pv. tomato strain DC3000. We excluded all the proteins related to T3SS, as well as the hypothetical proteins. (Note that this set may still contain some unknown effectors.) And then we selected randomly from the remaining samples to constitute the negative set, since if we use all of them, the data set would be too much imbalanced. The numbers of the data sets are listed in Table 1.

3.2. Experimental Settings and Evaluation Criteria. The classifier is built using the state-of-the-art supervised learning machinery, the SVM, which is widely used in bioinformatics. Our implementation of the SVM adopted LibSVM version 2.8 [17]. We considered polynomial, sigmoid, and RBF kernels for the SVM and observed that the RBF kernel has the best classification accuracy.

We used LDA model in the Matlab Topic Modeling Toolbox 1.4 [18]. As in LDA, the number of topics has great impact on its performance. The optimum number of topics was searched as described in Section 3.3. The other parameters used in the LDA model are set as follows: $\beta = 0.01$, $\alpha = 50/T$, where T is the number of topics, and the number of iterations is 500. The threshold m is set to be 40 according to the statistics of word occurrence times.

Multiple measures were used to assess the performance of our proposed method, including sensitivity, specificity, and total accuracy (TA). The sensitivity and specificity can be defined in terms of the number of true positives (TPs), the number of false positives (FPs), the number of false negatives (FNs) and the number of true negatives (TNs) as follows. We define

$$\begin{aligned} \text{Sensitivity} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{Specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}}. \end{aligned} \tag{7}$$

These two measures examine the ability of the correct classification for positive and negative samples, respectively. TA is the ratio of the samples classified correctly compared to the total size of the data set, which is calculated as follows:

$$\text{TA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \tag{8}$$

Considering that the maximal secretion or translocation may require the first 100 amino acids [19–21], in our experiments, the first 100 amino acids were used.

3.3. Number of Topics. The number of topics is a key parameter in LDA model because it directly influences the performance of the model. The perplexity is frequently used to assess the performance of LDA models. It measures the performance of the model, which is defined by [11]:

$$\text{perplexity}\,(D_{\text{test}}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d}\right\}. \tag{9}$$

This measure decreases monotonically in the likelihood of the test data; thus lower values indicate better modeling performance.

We calculated the values of perplexity on a held-out dataset. Figure 2 shows the perplexity plotted against the number of hidden topics, from 5 to 100. It can be observed that the perplexity decreases with an increasing number of topics. From 5 topics to 40 topics, the perplexity drops rapidly. When the number of topics is bigger than 40, the perplexity is almost constant. In our experiment, we set the number of topics to be 50.

3.4. Experimental Results. We have conducted a series of experiments to examine the performance of these two feature reduction methods and compared them with four other methods. Table 2 lists the number of dimension, total accuracy (TA), sensitivity, and specificity of these six methods. The method abbreviations and their corresponding description are in the following:

(1) dimer: using all the dimers without feature reduction;

(2) trimer: using all the trimers without feature reduction;

Table 2: Result comparison.

Method	Dimension	TA (%)	Sensitivity (%)	Specificity (%)
Dimer	400	94.2	91.4	94.5
Trimer	8000	90.4	100.0	90.2
Frequency	220	95.3	92.4	95.6
tf-idf	220	94.7	88.8	95.3
FRI	50	91.2	83.3	91.7
FRII	184	95.0	94.5	95.1

(3) frequency: using the dictionary words selected by word frequency;

(4) tf-idf: using the dictionary words selected by tf-idf value;

(5) FRI: using topic information as features;

(6) FRII: using the feature set which is based on (3) but further condensed by topic information.

From Table 2, we can find that all the six methods obtain total accuracies over 90%, which indicates that the amino acid patterns are competent for discriminating effectors and noneffectors.

In this study, long K-mers have little advantage for the classification. The dimer method has better performance than tri-mer method. Although the trimer method obtains a sensitivity of 100%, its total accuracy and specificity are much lower than other methods. That is because its false positive rate is very high. Since the prediction system aims to provide a reliable prediction result of effector candidates, the high false positive rate is not allowed.

Basically, all the new methods have satisfying performance. The feature selection methods using the dictionary words selected by frequency or tf-idf value achieve the best specificities and have overall better performance than the original dimer and trimer methods. It demonstrates that the strategy of dictionary construction and segmentation is successful in the protein sequence classification. The numbers of dimensions of these two methods are 220, including 20 amino acids, 150 dimers, and 50 trimers. The measure of frequency is better than tf-idf value, since the latter has a lower sensitivity. That may because the tf-idf value prefers to select some unusual words, which are not helpful for classification.

Obviously, the proposed feature reduction method I (FRI) has the smallest number of dimensions, but its accuracy is relatively low. FRII has 184 dimensions, including 20 single amino acids, 137 dimers, and 27 trimers. More trimers are discarded than dimers, because the frequencies of trimers are much lower and only a few of them can pass the criterion of word selection in Section 2.3. Actually, more trimers cannot improve the accuracy as we have mentioned before.

FRII achieves good results, even better than using all the dictionary words. The sensitivity of FRII is 2% higher than that of the frequency method, and the total accuracy and specificity are also comparable or better than other methods. These results indicate that although the topic space is not enough for the classification, the latent topic information is effective in selecting features.

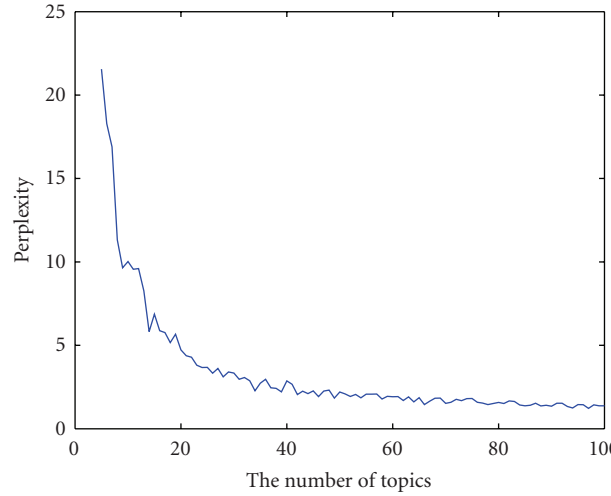


Figure 2: Perplexity under different numbers of topics.

4. Conclusions

This paper focuses on the feature reduction methods for identifying proteins secreted via the type III secretion system using machine learning approaches. Our goal is to extract features from N-terminal amino acid sequences and use the classifier to discriminate the input feature vectors as secreted or nonsecreted proteins.

We have compared six methods including the K-mer methods without feature reduction and other methods with different feature reduction approaches. Computational experiments were conducted on *Pseudomonas syringae* data set. The cross-validation tests on the *P. syringae* data set show that our methods achieve high accuracies.

We observe that, while long K-mer features have little contribution in discriminating effectors and noneffectors, conducting feature reduction can improve the prediction accuracy. The methods using frequency and tf-idf value for word selection achieve better accuracies than K-mer methods, and the further feature selection using topic information can improve the performance and condense the feature space at the same time.

Thus far, a large portion of T3SEs in Gram-negative bacteria still remain unknown. The bioinformatics tools are

of great importance. We believe that the new computational methods will contribute to the identification of novel type III secreted effectors and advance our understanding on TTSS.

As for the future work, the latent semantic information revealed by the topic models will be further investigated. LDA introduces a latent layer, which represents topic/subject in documents, or scene in images. For protein sequences, the latent layer could be secondary or spatial structure, function domain, or other biochemical properties. Since it is not as easy as images for proteins to visualize the sequences after running LDA, it is hard to define the specific corresponding concept of latent topic in the protein sequences. We will keep exploring the connection between biological characteristics and topics and incorporate other available information to discover the underlying mechanisms of the secretion system.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61003093) and the Science and Technology Program of Shanghai Maritime University (Grant no. 20110009).

References

[1] J. E. Galán and A. Collmer, "Type III secretion machines: bacterial devices for protein delivery into host cells," *Science*, vol. 284, no. 5418, pp. 1322–1328, 1999.

[2] S. Y. He, K. Nomura, and T. S. Whittam, "Type III protein secretion mechanism in mammalian and plant pathogens," *Biochimica et Biophysica Acta*, vol. 1694, no. 1–3, pp. 181–206, 2004.

[3] R. Arnold, S. Brandmaier, F. Kleine et al., "Sequence-based prediction of type III secreted proteins," *PLoS Pathogens*, vol. 5, no. 4, Article ID e1000376, 2009.

[4] Y. Yang, "A comparative study on sequence feature extraction for type III secreted effector prediction," in *Proceedings of the 8th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '11)*, July 2011.

[5] Y. Wang, Q. Zhang, M. A. Sun, and D. Guo, "High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles," *Bioinformatics*, vol. 27, no. 6, pp. 777–784, 2011.

[6] A. O. Ferreira, C. R. Myers, J. S. Gordon et al., "Whole-genome expression profiling defines the HrpL regulon of *Pseudomonas syringae* pv. tomato DC3000, allows de novo reconstruction of the Hrp cis clement, and identifies novel coregulated genes," *Molecular Plant-Microbe Interactions*, vol. 19, no. 11, pp. 1167–1179, 2006.

[7] E. M. Panina, S. Mattoo, N. Griffith, N. A. Kozak, M. H. Yuk, and J. F. Miller, "A genome-wide screen identifies a *Bordetella* type III secretion effector and candidate effectors in other species," *Molecular Microbiology*, vol. 58, no. 1, pp. 267–279, 2005.

[8] Y. Sato, A. Takaya, and T. Yamamoto, "Meta-analytic approach to the accurate prediction of secreted virulence effectors in gram-negative bacteria," *BMC Bioinformatics*, vol. 12, no. 1, article 442, 2011.

[9] Y. Yang, J. Zhao, R. L. Morgan, W. Ma, and T. Jiang, "Computational prediction of type III secreted proteins from gram-negative bacteria," *BMC Bioinformatics*, vol. 11, supplement 1, article S47, 2010.

[10] D. S. Guttman, B. A. Vinatzer, S. F. Sarkar, M. V. Ranall, G. Kettler, and J. T. Greenberg, "A functional screen for the type III (Hrp) secretome of the plant pathogen *Pseudomonas syringae*," *Science*, vol. 295, no. 5560, pp. 1722–1726, 2002.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[12] C. Wang, D. Blei, and L. Fei-Fei, "Simultaneous image classification and annotation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 1903–1910, June 2009.

[13] H. Zhang, B. Qiu, C. Giles, H. Foley, and J. Yen, "An LDA-based community structure discovery approach for large-scale social networks," in *Proceedings of the IEEE Intelligence and Security Informatics (ISI '07)*, pp. 200–207, May 2007.

[14] R. M. Neal, "Markov chain sampling methods for dirichiet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[15] D. Blei and M. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121–144, 2006.

[16] Y. Yang and B. L. Lu, "Extracting features from protein sequences using Chinese segmentation techniques for subcellular localization," in *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB '05)*, pp. 288–295, November 2005.

[17] C. C. Chang and C. J. Lin, 2001, LIBSVM: a library for support vector machines, software, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[18] M. Steyvers and T. Griffiths, 2011, Matlab Topic Modeling Toolbox 1.4, software, http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm.

[19] C. Casper-Lindley, D. Dahlbeck, E. T. Clark, and B. J. Staskawicz, "Direct biochemical evidencefor type III secretion-dependent translocation of the AvrBs2 effector protein into plant cells," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 8336–8341, 2002.

[20] T. Petnicki-Ocwieja, D. J. Schneider, V. C. Tam et al., "Genomewide identification of proteins secreted by the Hrp type III protein secretion system of *Pseudomonas syringae* pv. tomato DC3000," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 11, pp. 7652–7657, 2002.

[21] L. M. Schechter, K. A. Roberts, Y. Jamir, J. R. Alfano, and A. Collmer, "*Pseudomonas syringae* type III secretion system targeting signals and novel effectors studied with a Cya translocation reporter," *Journal of Bacteriology*, vol. 186, no. 2, pp. 543–555, 2004.

Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2012, Article ID 127130, 9 pages
doi:10.1155/2012/127130

Research Article

Prediction of Breeding Values for Dairy Cattle Using Artificial Neural Networks and Neuro-Fuzzy Systems

Saleh Shahinfar,[1] Hassan Mehrabani-Yeganeh,[1] Caro Lucas,[2] Ahmad Kalhor,[2] Majid Kazemian,[2] and Kent A. Weigel[3]

[1] *Department of Animal Science, University College of Agriculture and Natural Resources, University of Tehran, Karaj, Iran*
[2] *Center of Excellence: Control and Intelligent Processing, School of Electrical and Computer Engineering, University of Tehran, Iran*
[3] *Department of Dairy Science, University of Wisconsin-Madison, Madison, WI 53706, USA*

Correspondence should be addressed to Saleh Shahinfar, shahinfar@wisc.edu

Received 12 May 2012; Revised 9 August 2012; Accepted 9 August 2012

Academic Editor: Chunmei Liu

Developing machine learning and soft computing techniques has provided many opportunities for researchers to establish new analytical methods in different areas of science. The objective of this study is to investigate the potential of two types of intelligent learning methods, artificial neural networks and neuro-fuzzy systems, in order to estimate breeding values (EBV) of Iranian dairy cattle. Initially, the breeding values of lactating Holstein cows for milk and fat yield were estimated using conventional best linear unbiased prediction (BLUP) with an animal model. Once that was established, a multilayer perceptron was used to build ANN to predict breeding values from the performance data of selection candidates. Subsequently, fuzzy logic was used to form an NFS, a hybrid intelligent system that was implemented via a local linear model tree algorithm. For milk yield the correlations between EBV and EBV predicted by the ANN and NFS were 0.92 and 0.93, respectively. Corresponding correlations for fat yield were 0.93 and 0.93, respectively. Correlations between multitrait predictions of EBVs for milk and fat yield when predicted simultaneously by ANN were 0.93 and 0.93, respectively, whereas corresponding correlations with reference EBV for multitrait NFS were 0.94 and 0.95, respectively, for milk and fat production.

1. Introduction

Machine learning techniques, such as decision trees and artificial neural networks (ANN), are used increasingly in agriculture, because they are quick, powerful, and flexible tools for classification and prediction applications, particularly those involving nonlinear systems [1]. These techniques have been used for detection of mastitis [2], detection of estrus [3], and discovery of reasons for culling [1]. Decision trees and related methods have also been used for analysis of lactation curves [4], interpretation of somatic cell count data [5], and assessment of the efficiency of reproductive management [6, 7]. In addition, ANN have been used for the prediction of total farm milk production [8], prediction of 305-day milk yield [9, 10], and detection of mastitis [11, 12].

Fuzzy logic, which involves classification of variables into fuzzy sets with degrees of membership between 0 and 1, has recently found its way into agricultural research [13, 14]. Applications have included development of decision-support systems for analyzing test-day milk yield data from Dairy Herd Improvement (DHI) programs [15]. Detection of mastitis and estrus from automated milking systems [16, 17], and definition of contemporary groups for the purpose of genetic evaluation [18]. A key challenge in the use of fuzzy sets is the development of appropriate membership functions (MF). Due to the relative simplicity of building ANN, these may be used to reduce the time and computational burden associated with MF determination. In fact, developments in neural network-driven fuzzy control suggest that these technologies may be quite complementary

[19]. In multivariate prediction models, ANN could be used to develop MF for fuzzy sets from input variables such as milk yield, parity, or stage of lactation. Such tools have been used to develop decision-support software for culling and replacement decisions [20], as well as for qualitative assessment of milk production [21] in dairy cattle.

From a dairy cattle breeding viewpoint, accurate and timely prediction of lactation milk yield of progeny is a key prerequisite to selection of genetically superior males. In a breeding program, genetic progress can be maximized through accurate identification of superior animals that will be selected as parents of the next generation and therefore breeding goals can be achieved. A key component of this process is fast and reliable prediction of breeding values for selection candidates. However, prediction of breeding values is often a computationally challenging and time consuming task, and therefore it is undertaken only periodically (e.g., quarterly or semiannually) in most countries. Rapid, low-cost alternatives that can provide approximate predictions of breeding values with acceptable accuracy could allow more timely selection and culling decisions by breeding companies or dairy producers. Rapid identification of superior males can lead to earlier collection and distribution of semen and more rapid genetic progress [21, 22]. In several studies, back-propagation algorithms have been used to develop ANN for the prediction of 305-day milk, fat, and protein [21, 23]. However, there has not been any published research into the application of neuro-fuzzy networks and ANNs in the prediction of EBV in dairy cows.

The objective of this current study was to investigate the potential of a hybrid intelligent system that combines artificial neural networks and fuzzy logic, also known as a neuro-fuzzy system (NFS) or neuro-fuzzy network (NFN) in order to compute the breeding values of Holstein cows for milk and fat production based on their performance data and EBV of their parents.

2. Material

2.1. Data Collection and Preprocessing. Data were provided by the Animal Breeding Center of Iran (ABCI, Tehran) and consisted of 119,899 lactation records of first parity Holstein cows that calved between 22 and 36 mo of age during the time period from 1990 to 2005. Milk and fat yield records were pre-adjusted for milking frequency. Data regarding environmental conditions included ambient temperature, ambient humidity, and length of the photoperiod. Herds included in the present study were representative of large commercial dairy herds in Iran. In Iran, milk production involves traditional dairy farms, which consist of roughly 6 million native and crossbred dairy cattle, and large commercial farms, which consist of approximately 0.8 million Holstein cows. Herds in the latter group practice intensive management and feed a total mixed ration containing concentrates (corn grain, soybean meal, fish meal, cotton seed, cottonseed meal, barley grain, canola meal, beet pulp, fat powder, vitamins, and minerals), alfalfa, and corn silage. These herds contain up to 3000 lactating cows, which are housed in free stalls and milked in parlor systems. Average milk production in these herds ranges from 8000 to 10,000 kg per 305 d lactation, and this exceeds the national average of about 6300 kg for Iranian Holsteins. Artificial insemination is used in more than 75% of the herds, and most cows are inseminated with imported semen or semen provided by ABCI. Generally speaking, ABCI is responsible for recording data, computing genetic evaluations, and establishing breeding strategies. Data regarding production traits (milk, fat, and protein yield), functional traits (longevity, calving ease, somatic cell count, and female fertility), and physical conformation traits are recorded on the majority of large commercial farms, and EBV for milk and fat yield and fat percentage are provided to farmers twice a year.

2.2. Genetic Analysis. Data were prepared using Visual FoxPro v6.0 (Microsoft, Redmond, WA, USA). The following multiple-trait animal model for best linear unbiased prediction (BLUP) was used to compute EBV for milk and fat yield:

$$y_{ijkl} = \mathrm{HYS}_i + \mathrm{Age}_j + \mathrm{DIM}_k + a_l + e_{ijkl}, \quad (1)$$

where y_{ijkl} = milk or fat yield observation on the lth animal in the kth level of days in milk, jth level of age at calving, and ith herd-year-season class, HYS_i = fixed effect of the ith herd-year-season class, Age_j = fixed effect of jth level of age at first calving, DIM_k = fixed effect of kth level of days in milk, a_l = random genetic effect of lth animal, distributed as $\mathbf{A}\sigma_a^2$, and e_{ijkl} is a random residual effect, distributed as $\mathbf{I}\sigma_e^2$. The genetic analysis was implemented using the PEST software [24]. In subsequent analyses using ANN and NFS, the sire and dam EBV from this genetic analysis were used as input variables, whereas the individual cow EBV were used as reference EBV for calculation of predictive ability of the networks.

2.3. Artificial Neural Networks. An artificial neural network, or ANN, is often simply referred to as a neural network, and it represents a nonlinear statistical modeling tool that is based on the concept of a biological neural network. Information flows through the network during the learning phase, and the ANN adapts its structure in order to model complex relationships between the input and output variables. The ANN consists of basic units, termed neurons, whose design is suggested by their biological counterparts. These artificial neurons have input paths, just as biological neurons have dendrites, and they have output paths, just as biological neurons have axons [9]. Both artificial and biological neurons also have predispositions (biases) that affect the strength of their output. The neuron combines the inputs, incorporates effects of the predisposition, and outputs signals. Learning occurs in both real and artificial neurons, and this alters the strength of connections between the neurons and the biases [25].

The training of ANN often facilitates discovery of previously unknown relationships between input and output variables, and these relationships have been used successfully in both classification and prediction problems [26]. Recognition of patterns in ANN occurs through training with data

samples and is independent of the particular form of the information [27]. However, the pattern recognition ability of networks can be improved by various techniques. Common approaches to improving network performance include: finding an optimum network architecture, determining an appropriate number of training cycles, varying the combinations of input variables [28], customizing the values of learning parameters [23], and preselecting or preprocessing of the data [27].

In this research, we used a feed forward backpropagation multilayer perceptron (MLP herein) algorithm. We used a four layer MLP containing 1 input layer, 2 hidden layers, and 1 output layer. Each node in the input layer corresponds to one explanatory variable. Nodes in the hidden layer contain hyperbolic tangent activation functions [29], $h = (e^{y_i} - e^{-y_i})/(e^{y_i} + e^{-y_i})$ and they take a weighted sum of all input variables, $y_i = \sum_j \omega_{ji}\chi_i$, where χ_i is an input variable and ω_{ji} is corresponding weight in layer j. Similarly, the output node(s) take a weighted sum of all nodes in second hidden layer and use the same activation function to calculate the output value. Learning (updating weights) in the backpropagation algorithm starts by summing the errors over all of the network output unit(s). For each output unit k, the error term is $E_k = o_k(1 - o_k)\ (t_k - o_k)$, where t_k and o_k are target and output for kth output of dth training example, respectively. Then, for each hidden layer the error term will be $E_h = o_h(1 - o_h)\sum_{k\in\text{output}} \omega_{kh}E_k$, where o_h is the output of the hidden layer and ω_{kh} is the weight of kth output neuron. Eventually, for updating each weight in the network we use $\omega_{ji} = \omega_{ji} + \Delta\omega_{ji}$ whit $\Delta\omega_{ji} = \eta E_j x_{ji}$, where η is called the learning rate (e.g., 0.05), E_j is the error term for the jth node, and x_{ji} is the input value for jth node in ith layer to which the weight is applied [30].

The tangent hyperbolic function also ranges from -1 to 1 and is differentiable, which has two advantages. First, it is necessary when using in backpropagation algorithm and second it gives a prediction range between -1 and 1 which is well suited for this study, because in our case, breeding values can take both positive and negative values.

2.4. Fuzzy Logic. Fuzzy logic is a form of multivalued logic that deals with approximate (rather than precise) reasoning and multiple truth values (rather than simply true and false). It involves the use of fuzzy sets, comprised of various categories that are expressed qualitatively by an expert, to which an element could partially belong. The degree to which an element belongs to a fuzzy set is defined by a membership function, or MF. For example, the milk production records of individual cows could be classified by an expert as very low, low, medium, high or very high. These categories would be represented by five fuzzy sets, and the record of a specific cow might belong partially to each of two adjacent sets, such as very low and low. When using a 100% membership scale, the expert may infer that a milk production record of 7134 kg belongs 90% to the very low set and 10% to the low set [31]. The functions that define the degrees of membership for specific values of the independent variable (in this case, milk production) are known as membership functions.

2.5. Neuro-Fuzzy Systems. Neuro-fuzzy systems, or NFS, are hybrid intelligent systems that combine the subjective reasoning features of fuzzy logic with the learning structure of neural networks. As such, NFS represent fuzzy logic models that are partially designed from expert knowledge and partially learned from the data. The close linkage between fuzzy logic models and neural networks motivated this data-driven approach to fuzzy modeling. Typically, the fuzzy logic model is represented in the structure of a neural network, and machine learning methods that have been established in a neural network context are applied to the NFS. The contemporary viewpoint is that fuzzy models can be learned directly from data, without first being drawn in a neural network structure, and some learning methods that are applied have no relationship to ANN. Nevertheless, the original terms of fuzzy neural network and neuro-fuzzy system have persisted for all types of fuzzy models that are learned from data [32].

The fundamental approach with locally linear neuro-fuzzy models (LLNF) is to divide the input space into small linear subspaces with fuzzy validity functions [32, 33]. Each linear part, along with its validity function which is a normalized Gaussian function, can be described as a fuzzy neuron. Thus, the total model is a NFS with one hidden layer, and a linear neuron in the output layer that simply computes the weighted sum of the outputs of locally linear neurons. Global optimization of the linear consequent parameters is carried out by least-squares [32]. An incremental tree-based learning algorithm, known as the locally linear model tree (LOLIMOT), can be used to tune the rule premise parameters, that is, to determine the validation hypercube for each locally linear model. At each iteration, the worst-performing locally linear neuron is designated to be divided. This learning algorithm provides an optimal linear or nonlinear model with maximum generalization, and it performs well in prediction applications. Only one parameter, the embedding dimension, must be specified before implementing the algorithm.

The structure and behavior of the local linear model tree algorithm is shown in Figures 1(a) and 1(b), respectively. The output of each local linear model is calculated by $\hat{y}_i = \omega_{i0} + \omega_{i1}u_1 + \cdots + \omega_{ip}u_p$, where $u = [u_1, u_2, \ldots, u_p]$ is the vector of inputs, ω_i s are linear coefficients, and a linear layer in the output simply calculates the weighted sum of each neuron (here local linear model outputs) as follows:

$$\hat{y} = \sum_{i=1}^{M} \hat{y}_i \phi_i(u), \tag{2}$$

where $\phi_i(u)$ is the validity function of each neuron, calculated as $\phi_i(u) = \mu_i(u)/\sum_{j=1}^{M}\mu_j(u)$

$$\mu_i(u) = \exp\left\{-\frac{1}{2}\left[\frac{(u_i - c_{i1})^2}{\sigma_{i1}^2} + \cdots + \frac{\left(u_p - c_{p1}\right)^2}{\sigma_{p1}^2}\right]\right\}$$

$$= \prod_{j=1}^{p} \exp\left[-\frac{1}{2} \frac{\left(u_j - c_{ij}\right)^2}{\sigma_{ij}^2} \right]. \tag{3}$$

The best advantage of LOLIMOT is its low computational time, which is linear with respect to the number of fuzzy neurons [32].

2.6. Network Development and Error Criteria. In the implementation of ANN methodology for predicting EBV of Iranian Holstein cows for milk and fat yield, several ANN were designed using forward stepwise selection of input variables which means that we started with only one input variable (Milk 2x) and tried to predict EBV for milk. Then we kept adding other variables to the input vector until we reached the full set of available input variables, as shown in Table 1. Only the best-performing networks were selected based on RMSE and the correlations between predicted output and actual EBV and are discussed herein. This trial and error approach for "tuning" the network is necessary to determine the optimum structure and appropriate parameters. These networks, which were developed using the multilayer perceptron (MLP) algorithm, have two hidden layers and tangent hyperbolic activation functions. In the first hidden layer, the number of nodes was chosen to be twice the number of input variables, and in second hidden layer, the number of nodes was chosen to be equal to the number of inputs. Three learning rules were considered when training the networks: momentum, conjugate gradient [34], and Levenberg [35]. However, because results were nearly identical for all three learning rules, only results from the conjugate gradient approach are discussed herein. These algorithms were implemented using NeuroSolutions v5.0 (NeuroDimensions, Gainesville, FL, USA). The maximum number of epochs was set to 5000. Each network was trained five times with different initial random weights, and the best weight was chosen for testing each network.

A "training set," which consisted of a random sample of 7000 observations from the full data set, was used for initial development and each the network. Subsequently, a "tuning set," which consisted of a random sample of 1000 additional observations, was used to optimize the structure of the network and determine appropriate parameter values via cross-validation. Finally, a "testing set," which consisted of a random sample of 2000 independent observations, was used to validate the performance of the network via testing. The error criteria used to evaluate network performance included root mean square error (RMSE) and the correlation between predicted and reference EBV (r), where reference EBV correspond to animal model BLUP EBVs from the genetic analysis described in an earlier section:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{\sum_i (x_i - d_i)^2}{n}} \\ r &= \frac{\sum_i (x_i - \overline{x})\left(d_i - \overline{d}\right)/(n-1)}{\sqrt{\sum_i (d_i - \overline{d})^2/(n-1)}\sqrt{\sum_i (x_i - \overline{x})^2/(n-1)}}, \end{aligned} \tag{4}$$

where n = number of observations in the data set; d_i = reference output for observation i, with mean $\overline{d}$; x_i = predicted output for observation i, with mean $\overline{x}$.

The data set described above was also used to train and evaluate neuro-fuzzy systems with the locally linear model tree algorithm, or LOLIMOT, using MATLAB v7.0 (The MathWorks, Natick, MA, USA). In these networks, prediction of desired outputs progressed until 50 locally linear models were developed. Subsequently, the best model was chosen according to the RMSE error criterion, and this model was used for validation of the network in the testing set.

3. Results and Discussion

A total of 20 networks were evaluated for each multilayer perceptron and locally linear model tree algorithm. The input variables considered in each of the 20 networks are presented in Table 1, and these include: age at first calving, days in milk, ambient temperature, ambient humidity, length of the photoperiod, raw and adjusted (for milking frequency) milk, and fat production of each cow and the average of her contemporaries, and the milk and fat EBV of her parents. The output variables included: single-trait milk EBV of the cow (networks 1 to 13), single-trait fat EBV of the cow (networks 14 and 15), and multitrait milk and fat EBV of the cow (networks 16 to 20).

3.1. Prediction of EBV for Milk Yield. In experiments 1 to 3, milk yield EBV were predicted as a single trait, using ANN via the MLP algorithm and NFS via the LOLIMOT algorithm as a function of fix effects, milk production, environmental factors and milk yield EBV of the dams. Results are given in Table 2. As the number of input variables increased in networks 1 to 3, the correlation between actual and predicted EBV increased, and the error criteria decreased, specially with regard to RMSE. In experiments 1 and 2, NFS had better performance than the ANN, whereas in experiment 3 performance of ANN was slightly superior.

In experiments 4 to 6, milk yield EBV of the dams were not considered in the vector of input variables, but milk yield EBV of the sires were included. This resulted in a substantial decrease in the correlation between reference EBV and EBV predicted by the ANN and NFS, suggesting that EBV of the dam is a more useful variable for prediction of EBV of her daughters. This was an unexpected result given that sires' EBV are generally more accurate than dams' EBV, and it was most likely due to a common environmental component between a cow and her dam (note that herd-year-season was not included in the ANN and NFS because this variable would have had explanatory power in the training set, while providing no predictive power in the testing set, even for future observations in the same herds).

In experiments 7 to 9, environmental variables were considered along with the EBV of both the cow's sire and the cow's dam. This resulted in higher correlations, as compared with experiments 1 to 3 and experiments 4 to 6, most notably the latter. This suggests that both sire and dam EBV can be

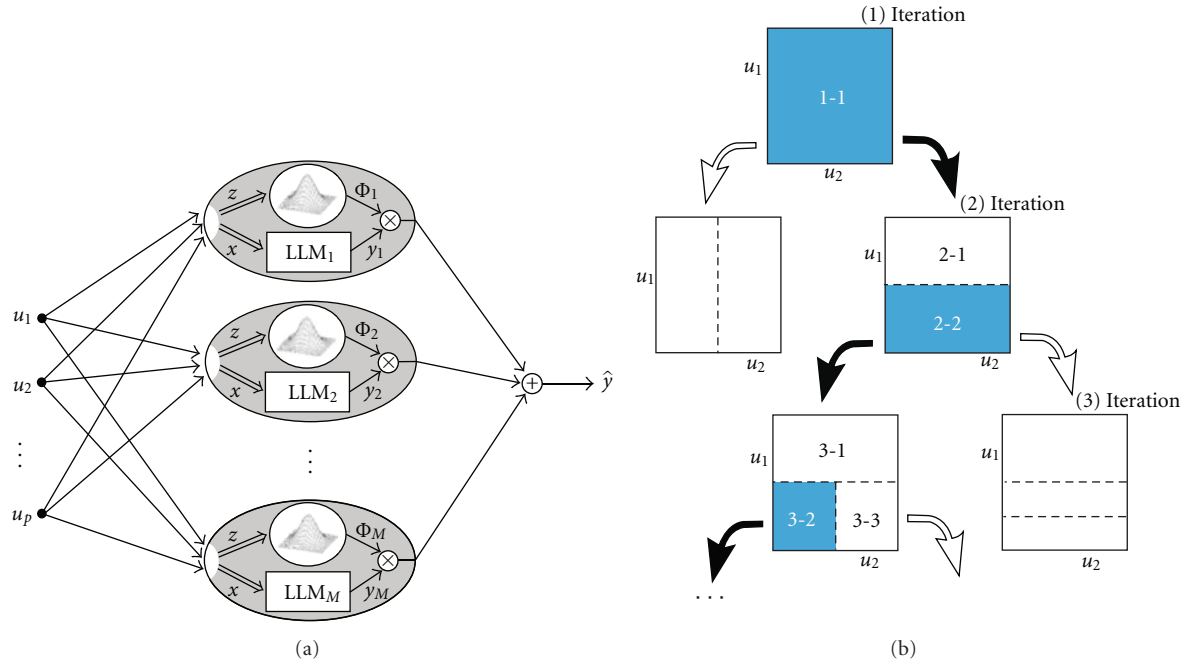


Figure 1: (a) Description of LOLIMOT Architecture, (b) description of LOLIMOT algorithm.

Table 1: Inputs and outputs of various twenty experiments in this study.

Experiment no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Inputs																				
Age	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Days in milk		*	*		*	*		*	*	*	*	*	*	*	*	*	*	*	*	*
Milk 2x	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Fat 2x										*	*	*	*	*	*	*	*	*	*	*
Herd mean milk 2x										*	*	*	*	*	*	*	*	*	*	*
Herd mean fat 2x										*	*	*	*	*	*	*	*	*	*	*
Herd mean milk total																				*
Total milk																	*		*	*
Temperature			*			*			*	*	*	*	*	*	*	*	*	*	*	*
Humidity										*	*	*	*	*	*	*	*	*	*	*
Day length										*	*	*	*	*	*	*	*	*	*	*
Milk EBV of dam	*	*	*				*	*	*	*	*	*	*	*	*	*	*	*	*	*
Fat EBV of dam											*		*	*	*	*	*	*	*	*
Milk EBV of sire				*	*	*	*	*	*			*	*					*	*	*
Fat EBV of sire															*			*	*	*
Outputs																				
Milk EBV	*	*	*	*	*	*	*	*	*	*	*	*	*			*	*	*	*	*
Fat EBV														*	*	*	*	*	*	*

useful predictors of the EBV of their offspring, as one would expect, but that the dam's EBV provides more information in this type of analysis for the reason noted previously. In experiments 7 and 8, correlations and RMSE criteria indicated greater predictive ability for NFS, whereas in experiment 9, predictive ability was slightly greater for ANN.

In experiments 10 and 11, additional variables such as herd average, fat yield, humidity and length of day were included in the input vector. Predictive ability improved with inclusion of these variables; presumably most of this gain can be attributed to the inclusion of herd average. In experiment 11, fat yield EBV of the dam was also considered. Unexpectedly, addition of this variable improved performance of NFS slightly, but performance of ANN deteriorated, apparently because it introduced additional "noise" into the analysis. In experiments 12 and 13, milk yield EBV of the sire was also

Table 2: Mean square error, root mean square error, and correlation in thirteen MLP and neuro-fuzzy networks for predicting milk EBV.

Networks		MLP		LOLIMOT	
Error criteria		RMSE	r	RMSE	r
Experiment no.	1	192.3	0.69	184.022	0.81
	2	156.5	0.81	154.5	0.82
	3	149.8	0.83	153.4	0.83
	4	208.1	0.63	210.6	0.63
	5	212.0	0.61	206.8	0.66
	6	172.8	0.67	205.5	0.68
	7	154.1	0.82	144.2	0.82
	8	151.6	0.82	143.1	0.83
	9	144.3	0.85	143.4	0.84
	10	109.7	0.91	113.1	0.92
	11	117.9	0.90	113.0	0.92
	12	106.7	0.92	**101.8**	**0.93**
	13	**106.2**	**0.92**	102.0	0.93

Table 3: Mean square error, root mean square error, and correlation in two MLP and neuro-fuzzy networks for predicting fat EBV.

Networks		MLP		LOLIMOT	
Error criteria		RMSE	r	RMSE	r
Experiment no.	14	3.1	0.91	3.3	0.91
	15	**2.7**	**0.93**	**2.8**	**0.93**

added to the input vector. In this case, inclusion of the dam's EBV for fat yield provided a very small increase in predictive ability of ANN and a very slight decrease in predictive ability of NFS, suggesting that dam's EBV for fat yield is largely redundant once milk yield EBV of the sire and dam and milk and fat yield of the cow have already been considered.

Overall, the predictive ability of the best networks for milk yield, namely, experiment 13 for ANN and experiment 12 for NFS, was outstanding. Correlations between predicted milk yield EBV from the ANN and NFS analyses and reference EBV from BLUP analysis of the full data set were 0.92 and 0.93, respectively.

3.2. Prediction of EBV for Fat Yield. In experiments 14 and 15, the objective was to predict fat yield EBV in a single trait analysis. As shown in Table 1, input variables were equivalent to those used for prediction of milk yield EBV in experiments 12 and 13 with fat EBV of Dam and Sire replacing milk EBV of Dam and Sire in those experiments. However, as shown in Table 3, performance was slightly better for prediction of EBV for fat than for milk, with correlations between ANN and NFS predictions and reference EBVs of 0.93 and 0.93, respectively, in experiment 15.

3.3. Simultaneous Prediction of Milk and Fat EBV. In experiments 16 to 20, the objective was to jointly predict EBVs for milk yield and fat yield in a single analysis. Total milk yield of the animal (i.e., beyond 305 d) was included as an input variable in experiments 17, 19, and 20, and herd average for total milk yield was also included in experiment 20. As shown in Table 4, the addition of total milk yield provided a slight improvement in predictive ability in the NFS, but performance of the ANN deteriorated slightly, perhaps indicating that the information provided by this variable was largely redundant. Experiment 18 was equivalent to the single-trait analyses (i.e., its input variables was union set of variables in experiments 13 and 15), and predictive ability was equal to or better than in the single-trait analyses. In both experiments 18 and 19, performance of the NFS was superior to that of the ANN. Lastly, in experiment 20, all available explanatory variables listed in Table 1 were used in the simultaneous prediction of EBV for milk and fat yield. As one might expect, this experiment provided the highest correlations with reference EBV and, in general, the smallest RMSE of prediction.

Figures 2 and 3 show the relationship between the number of neurons in the NFS analyses and the root mean square error of prediction in the training and testing sets. These figures clearly illustrate the danger of overfitting the training data. In every case, increasing model complexity (via the addition of more neurons) continuously improved predictive ability within the training set. However, predictive ability within the testing set, which is the true measure of expected performance in future, independent data sets, can be compromised by overfitting. In some cases (e.g., Figures 2(a) and 3(a)) the cost of overfitting was small, but in other cases (e.g., Figure 2(b)) performance in the testing set was significantly impaired by unnecessary increases in model complexity. In practice, users should monitor cross-validation predictive ability in the tuning set (i.e., a "set aside" portion of the training set) to avoid overfitting and thereby optimize parameters of the model.

4. Conclusions

The current methods for computing EBV, which involve simultaneous animal model BLUP analysis of all performance-recorded animals in the population, are computationally intensive and time-consuming. As such, EBVs are computed only periodically, usually two or three times per year. Therefore, it may be useful to develop an alternative approach for routine computation of EBV of dairy sires and cows, so that new data can be incorporated as soon as it becomes available. With this in mind, we evaluated the possibility of calculating approximate EBV using computationally efficient algorithms from the fields of artificial intelligence and machine learning, namely, artificial neural networks, or ANN, and neuro-fuzzy systems, or NFS. Using ANN and NFS approaches, we produced single trait predictions of milk yield EBV that had correlations of 0.917 and 0.926, respectively, and for fat yield EBV that had correlations of 0.926 and 0.932, respectively, with reference EBV. Furthermore, joint prediction of milk and fat yield EBV in multiple-trait implementations of ANN provided correlations of 0.925 and 0.930, respectively, with reference EBV for milk and fat production. The same prediction with

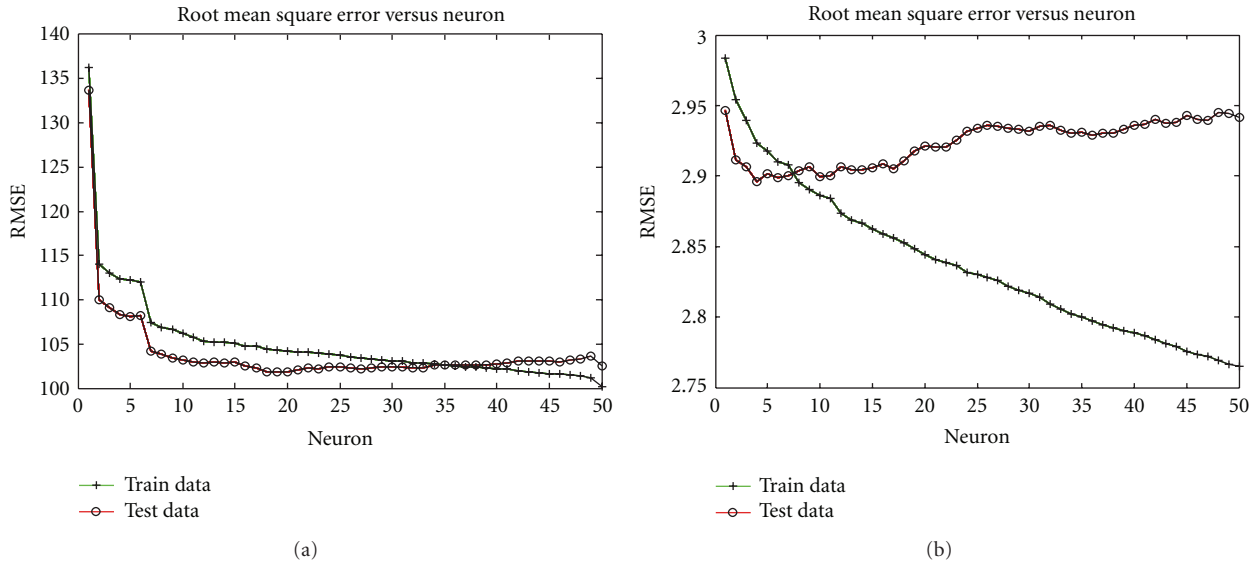


Figure 2: Root mean square error (RMSE) as a function of number of neurons in the single-trait neuro-fuzzy models: (a) prediction of milk yield EBV in experiment 12, and (b) prediction of fat yield EBV in experiment 15.

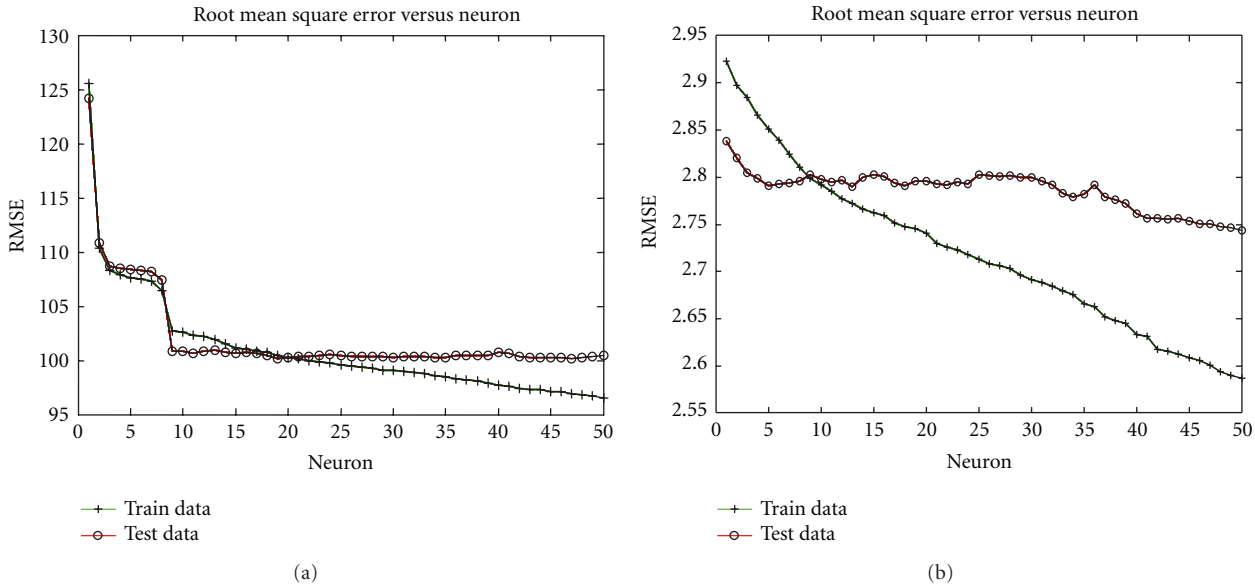


Figure 3: Root mean square error (RMSE) as a function of number of neurons in the multiple-trait neuro-fuzzy models: (a) prediction of milk yield EBV in experiment 20, and (b) prediction of fat yield EBV in experiment 20.

NFS provided a correlation of 0.935 and 0.949 with reference EBV, respectively, for milk and fat.

In most cases, NFS tended to provide greater predictive ability than ANN. However, the difference in performance between these two methods was rather small. For both methods, increasing the number of input variables led to predictions of EBV with greater accuracy. In general, however, the NFS approach seemed to provide slightly more consistent results, and this method may be more robust to "noise" in specific data sets or redundancies among specific combinations of explanatory variables. Some novel aspects of the NFS approach are advantageous as compared with conventional ANN methodology. For example, learning of model trees in the LOLIMOT algorithm leads to automatic adaptation of the complexity of the network structure to the requirements of a particular application. Considerable post-pruning would be required to achieve similar results using ANN. In addition, every neuron in the NFS implementation,

Table 4: Mean square error, root mean square error, and correlation in five MLP and neuro-fuzzy networks for predicting milk and fat EBV simultaneously.

Networks		MLP				LOLIMOT			
Trait		Milk		Fat		Milk		Fat	
Error criteria		RMSE	r	RMSE	r	RMSE	r	RMSE	r
Experiment no.	16	122.3	0.89	4.45	0.88	113.1	0.92	3.30	0.91
	17	117.7	0.90	4.33	0.88	113.7	0.92	3.32	0.91
	18	105.5	0.90	5.11	0.92	102.6	0.93	2.84	0.94
	19	103.8	0.92	5.07	0.92	102.4	0.93	2.77	0.94
	20	**101.4**	**0.93**	**4.93**	**0.93**	**100.2**	**0.94**	**2.75**	**0.95**

via the LOLIMOT algorithm, is a linear regressor, and therefore the resulting solution is much more transparent than that of an ANN. In addition to not having learning problems such as suboptimality due to local minima, it can provide a better explanation engine and means to use partial expert knowledge in the linear form.

Lastly, it must be emphasized that the application of such novel methods for computation of EBV in animal breeding is quite new, and as such a period of learning and adaptation will be required before such approaches can be implemented in an optimal manner.

Abbreviations

ANN: Artificial neural network
MLP: Multilayer perceptron
LOLIMOT: Locally linear model tree
LLNF: Locally linear neuro-fuzzy
NFS: Neuro fuzzy system
EBV: Estimated breeding value
MF: Membership function.

Acknowledgments

The authors wish to acknowledge the assistance of Mr. Jalal Fatehi from the Centre for Genetic Improvement of Livestock (Guelph, ON, Canada) in implementing the genetic evaluation software. Also we extend our appreciation to Adam Pergament for his constructing guide in editing the paper. S. Shahinfar wants to dedicate this work to C. Lucas, 3rd author of the paper who passed away before this work can be published.

References

[1] R. J. McQueen, S. R. Garner, C. G. Nevill-Manning, and I. H. Witten, "Applying machine learning to agricultural data," *Computers and Electronics in Agriculture*, vol. 12, no. 4, pp. 275–293, 1995.

[2] T. Kim and C. W. Heald, "Inducing inference rules for the classification of bovine mastitis," *Computers and Electronics in Agriculture*, vol. 23, no. 1, pp. 27–42, 1999.

[3] R. S. Mitchell, R. A. Sherlock, and L. A. Smith, "An investigation into the use of machine learning for determining oestrus in cows," *Computers and Electronics in Agriculture*, vol. 15, no. 3, pp. 195–213, 1996.

[4] D. Pietersma, R. Lacroix, D. Lefebvre, and K. M. Wade, "Induction and evaluation of decision trees for lactation curve analysis," *Computers and Electronics in Agriculture*, vol. 38, no. 1, pp. 19–32, 2003.

[5] H. G. Allore, L. R. Jones, W. G. Merrill, and P. A. Oltenacu, "A decision support system for evaluating mastitis information," *Journal of Dairy Science*, vol. 78, no. 6, pp. 1382–1398, 1995.

[6] J. J. Domecq, R. L. Nebel, M. L. McGilliard, and A. T. Pasquino, "Expert system for evaluation of reproductive performance and management," *Journal of Dairy Science*, vol. 74, no. 10, pp. 3446–3453, 1991.

[7] D. Z. Caraviello, K. A. Weigel, M. Craven et al., "Analysis of reproductive performance of lactating cows on large dairy farms using machine learning algorithms," *Journal of Dairy Science*, vol. 89, no. 12, pp. 4703–4722, 2006.

[8] L. Sanzogni and D. Kerr, "Milk production estimates using feed forward artificial neural networks," *Computers and Electronics in Agriculture*, vol. 32, no. 1, pp. 21–30, 2001.

[9] R. K. Sharma and A. K. Sharma, "Neuro-computing paradigms with application to dairy production," in *Lecture Compendium, National Training Programme on InFormation Resources on Genetics and Documentation Techniques For Livestock Improvement*, A. Sharma et al., Ed., pp. 173–178, Centre For Advanced Studies—AG&B, Dairy Cattle Breeding Division, NDRI, Karnal, India, 2004.

[10] S. Shahinfar, *Intelligent decision making for progeny testing in dairy cattle using neuro-fuzzy models [M.S. thesis]*, University of Tehran, Tehran, Iran, 2007.

[11] C. W. Heald, T. Kim, W. M. Sischo, J. B. Cooper, and D. R. Wolfgang, "A computerized mastitis decision aid using farm-based records: an artificial neural network approach," *Journal of Dairy Science*, vol. 83, no. 4, pp. 711–720, 2000.

[12] M. Nielen, Y. H. Schukken, A. Brand, S. Haring, and R. T. Ferwerda-van Zonneveld, "Comparison of analysis techniques for on-line detection of clinical mastitis," *Journal of Dairy Science*, vol. 78, no. 5, pp. 1050–1061, 1995.

[13] S. Kim and S. I. Cho, "Neural network modeling and fuzzy control simulation for bread-baking process," *Transactions of the ASAE*, vol. 40, pp. 671–676, 1997.

[14] R. S. F. Ribeiro, R. E. Yoder, J. B. Wilkerson, and B. D. Russell, "A fuzzy logic base control system optimized via neural networks," ASAE Paper 98:2169, 1998.

[15] R. Lacroix, J. Huijbers, R. Tiemessen, D. Lefebvre, D. Marchand, and K. M. Wade, "Fuzzy set-based analytical tools for dairy herd improvement," *Applied Engineering in Agriculture*, vol. 14, no. 1, pp. 79–85, 1998.

[16] R. M. De Mol and W. E. Woldt, "Application of fuzzy logic in automated cow status monitoring," *Journal of Dairy Science*, vol. 84, no. 2, pp. 400–410, 2001.

[17] C. Kamphuis, R. Sherlock, J. Jago, G. Mein, and H. Hogeveen, "Automatic detection of clinical mastitis is improved by in-line monitoring of somatic cell count," *Journal of Dairy Science*, vol. 91, no. 12, pp. 4560–4570, 2008.

[18] R. Rekaya, K. A. Weigel, and D. Gianola, "Comparison between traditional and fuzzy logic definitions of herd-year-season groups in Holsteins," *Journal of Animal Science*, vol. 77, supplement 1, p. 33, 1999.

[19] H. Takagi and I. Hyashi, "NN-Driven fuzzy reasoning," *International Journal of Approximate Reasoning*, vol. 5, pp. 91–212, 1991.

[20] R. Lacroix, M. Strasser, R. Kok, and K. M. Wade, "Performance analysis of a fuzzy decision-support system for culling of dairy cows," *Canadian Agricultural Engineering*, vol. 40, no. 2, pp. 139–151, 1998.

[21] F. Salehi, R. Lacroix, and K. M. Wade, "Improving dairy yield predictions through combined record classifiers and specialized artificial neural networks," *Computers and Electronics in Agriculture*, vol. 20, no. 3, pp. 199–213, 1998.

[22] R. Lacroix, K. M. Wade, R. Kok, and J. F. Hayes, "Prediction of cow performance with a connectionist model," *Transactions of the American Society of Agricultural Engineers*, vol. 38, no. 5, pp. 1573–1579, 1995.

[23] F. Salehi, R. Lacroix, and K. M. Wade, "Effects of learning parameters and data presentation on the performance of back-propagation networks for milk yield prediction," *Transactions of the American Society of Agricultural Engineers*, vol. 41, no. 1, pp. 253–259, 1998.

[24] E. Groeneveld, *PEST Users' Manual*, Institute of Animal Husbandry and Animal Behavior Federal Research Center, Neustadt, Germany, 1990.

[25] A. K. Sharma, R. K. Sharma, and H. S. Kasana, "Prediction of first lactation 305-day milk yield in Karan Fries dairy cattle using ANN modeling," *Applied Soft Computing Journal*, vol. 7, no. 3, pp. 1112–1120, 2007.

[26] M. S. Sayeed, A. D. Whittaker, and N. D. Kehtarnavaz, "Snack quality evaluation method based on image features and neural network prediction," *Transactions of the American Society of Agricultural Engineers*, vol. 38, no. 4, pp. 1239–1245, 1995.

[27] R. Ruan, S. Almaer, C. Zou, and P. L. Chen, "Spectrum analysis of mixing power curves for neural network prediction of dough rheological properties," *Transactions of the American Society of Agricultural Engineers*, vol. 40, no. 3, pp. 677–681, 1997.

[28] R. S. Parmar, R. W. McClendon, G. Hoogenboom, P. D. Blankenship, R. J. Cole, and J. W. Dorner, "Estimation of aflatoxin contamination in preharvest peanuts using neural networks," *Transactions of the American Society of Agricultural Engineers*, vol. 40, no. 3, pp. 809–813, 1997.

[29] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, PWS Publishing, Boston, Mass, USA, 1996.

[30] T. M. Mitchell, *Machine Learning*, McGraw-Hill, New York, NY, USA, 1997.

[31] F. Salehi, R. Lacroix, and K. M. Wade, "Development of neuro-fuzzifiers for qualitative analyses of milk yield," *Computers and Electronics in Agriculture*, vol. 28, no. 3, pp. 171–186, 2000.

[32] O. Nelles, *Nonlinear System Identification*, Springer, Berlin, Germany, 2001.

[33] O. Nelles, *Nonlinear system identification with local linear neuro-fuzzy models [Ph.D. thesis]*, TU Darmstadt, Shaker, Aachen, Germany, 1999.

[34] M. R. Hestenes and E. Stiefel, "Method of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 6, article 49, 1952.

[35] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2012, Article ID 320698, 7 pages
doi:10.1155/2012/320698

Research Article

A Novel Weighted Support Vector Machine Based on Particle Swarm Optimization for Gene Selection and Tumor Classification

Mohammad Javad Abdi,[1] Seyed Mohammad Hosseini,[1] and Mansoor Rezghi[2]

[1] *Department of Computer Sciences, Faculty of Mathematical Sciences, Tarbiat Modares University, P.O. Box 14115-134, Tehran, Iran*
[2] *Department of Applied Mathematics, Sahand University of Technology, Tabriz, Iran*

Correspondence should be addressed to Mohammad Javad Abdi, mj.abdi@modares.ac.ir

Received 14 February 2012; Revised 12 May 2012; Accepted 15 May 2012

Academic Editor: Dongsheng Che

We develop a detection model based on support vector machines (SVMs) and particle swarm optimization (PSO) for gene selection and tumor classification problems. The proposed model consists of two stages: first, the well-known minimum redundancy-maximum relevance (mRMR) method is applied to preselect genes that have the highest relevance with the target class and are maximally dissimilar to each other. Then, PSO is proposed to form a novel weighted SVM (WSVM) to classify samples. In this WSVM, PSO not only discards redundant genes, but also especially takes into account the degree of importance of each gene and assigns diverse weights to the different genes. We also use PSO to find appropriate kernel parameters since the choice of gene weights influences the optimal kernel parameters and vice versa. Experimental results show that the proposed mRMR-PSO-WSVM model achieves highest classification accuracy on two popular leukemia and colon gene expression datasets obtained from DNA microarrays. Therefore, we can conclude that our proposed method is very promising compared to the previously reported results.

1. Introduction

Microarray technology is a tool for analyzing gene expressions consisting of a small membrane containing samples of many genes arranged in a regular pattern. Microarrays may be used to assay gene expression within a single sample or to compare gene expression in two different cell types or tissue samples, such as in healthy and cancerous tissue. The use of this technology is increased in recent years to identify genes involved in the development of diseases. Various clustering, classification, and prediction techniques have been utilized to analyze, classify, and understand the gene expression data such as Fisher discriminant analysis [1], artificial neural networks [2], and support vector machines (SVM) [3]. Briefly, SVM is a supervised learning algorithm based on statistical learning theory introduced by Vapnik [4]. It has great performance since it can handle a nonlinear classification efficiently by mapping samples from low dimensional input space into high dimensional feature space with a nonlinear kernel function. It is useful in handling classification tasks for high-dimensional and sparse microarray data and has been recommended as an effective approach to treat this specific data structure [5–8]. Due to its many attractive characters, it has been also widely used in various fields such as image recognition, text classification, speaker identification, and medical diagnosis, bioinformatics. Therefore, our study intends to investigate the application of SVM in tumor classification problem and suggests an effective model to minimize its error rate.

It is well known that SVM assumed that all the available genes of certain gene expression data have equal weights in classification process. However, for a real tumor classification problem each gene may possess different relevance to the classification results. Thus, the genes with more relevance are more important than those with less relevance. Usually, there are two approaches to tackle this issue. One strategy is gene selection aiming at determination of a subset of genes which is most discriminative and informative for classification. The other is gene weighting which seeks to estimate the relative importance of each gene and assign it a corresponding weight [9–11]. Gene selection has attracted increasing interests in bioinformatics in recent years because

its results can effectively help cancer diagnosis and clinical treatment. In this case, many outstanding methods based on particle swarm optimization (PSO) have been developed. PSO is a new evolutionary computation technique proposed by Kennedy and Eberhart [12] which was motivated by simulations of bird flocking or fish schooling. Shen et al. [8] introduced a combination of PSO and support vector machines (SVMs) for gene selection and tumor classification problem. In their work, the modified discrete PSO was applied to select genes and SVM to diagnose colon tumor. They also proposed a combination of PSO and tabu search (TS) approaches for gene selection problem [13]. The combination of TS as a local improvement procedure and PSO enabled their algorithm to overleap local optima and showed satisfactory performance. In 2008, Chuang et al. [14] suggested an improved binary PSO. The main contribution of their work was resetting all the global best particle positions after no change in three consecutive iterations. Li et al. [15] introduced a novel hybrid of PSO and genetic algorithms (GA) for the same purpose, overcoming the local optimum problem.

On the other hand, instead of making a binary decision on a genes' relevance, gene weighting utilizes a continuous value and hence has a finer granularity in determining the relevance. The strategy proposed in this work is a combination of gene selection and gene weighting. The proposed method consists of two stages. First, we apply minimum redundancy-maximum relevance (mRMR) method, proposed by Hanchuan et al. [16], to preselect genes having the highest relevance with the target class and being maximally dissimilar to each other. Then, PSO is employed to form a novel weighted SVM (WSVM) to classify samples. In this WSVM, PSO not only discards redundant genes (gene selection), but also especially takes into account the degree of importance of each gene and assigns diverse weights to the different genes (gene weighting). To construct an accurate SVM, we also use PSO to find appropriate kernel parameters, since the choice of gene weights influences the optimal kernel parameters and vice versa. Experimental results show that our proposed method (called mRMR-PSO-WSVM) achieves higher classification rate than previously reported results.

The rest of this paper is organized as follows. The following section provides a brief description of the well-known mRMR filter method, SVM classifier, weighted SVM and PSO besides the proposed method, respectively. Experimental results and conclusions are demonstrated in Sections 3 and 4, respectively.

2. Method

2.1. Minimum Redundancy-Maximum Relevance (mRMR). In this work a well-designed filter method, mRMR, is employed to enhance the gene selection in achieving both high accuracy and fast speed. In high-dimensional microarray data, due to the existence of a set of several thousands of genes, it is hard and even infeasible for SVM to be trained accurately. Alternative methods should be effectively applied to tackle this problem. Therefore, first of all, mRMR is applied to filter noisy and redundant genes. More specifically, mRMR method [16] is a criterion for first-order incremental gene selection, which is warmly being studied by a great number of researchers. In mRMR, genes which have both minimum redundancy for input genes and maximum relevancy for disease classes should be selected. Thus this method is based on two important metrics. One is mutual information between disease classes and each gene, which is used to measure the relevancy, and the other is mutual information between every two genes, which is employed to compute the redundancy. Let S denote the subset of selected genes, and Ω is the set of all available genes; the minimum redundancy can be computed by

$$\min_{S\subset\Omega} \frac{1}{|S|^2} \sum_{i,j\in S} I\left(g_i, g_j\right), \tag{1}$$

where $I(g_i, g_j)$ is the mutual information between ith and jth genes which measures the mutual dependence of these two variables. Formally, the mutual information of two discrete random variables g_i and g_j can be defined as

$$I\left(g_i, g_j\right) = \sum_{m\in g_i} \sum_{n\in g_j} p(m,n) \log \frac{p(m,n)}{p(m)p(n)}, \tag{2}$$

where $p(m,n)$ is the joint probability distribution function of g_i and g_j, and $p(m)$ and $p(n)$ are the marginal probability distribution functions of g_i and g_j, respectively [17]. In (4), $|S|$ is the number of genes of S. In contrast, mutual information $I(T, g_j)$ is usually employed to calculate discrimination ability from gene g_i to class $T = \{t_1, t_2\}$, where t_1 and t_2 denote the healthy and tumor classes. Therefore, the maximum relevancy can be calculated by

$$\max_{S\subset\Omega} \frac{1}{|S|} \sum_{i\in S} I(T, g_i). \tag{3}$$

Combined (5) with (6), mRMR feature selection criterion can be obtained as below in difference form:

$$\max_{S\subset\Omega} \left\{ \sum_{i\in S} I(T, g_i) - \left[\frac{1}{|S|} \sum_{i,j\in S} I\left(g_i, g_j\right) \right] \right\}. \tag{4}$$

2.2. Support Vector Machines (SVM). SVM classifier is briefly described as follows [18, 19]. Assume $\{x_i, y_i\}_{i=1}^{N}$ is a training dataset, where x is the input sample, and $y \in \{+1, -1\}$ is the label of classes. The SVM aim is to determine a hyper plane that optimally separates two classes using training dataset. This hyper plane is defined as $w \cdot x + b = 0$, where x is a point lying on the hyper plane, w determines the orientation of the hyper plane, and b is the bias of the distance of hyper plane from the origin. To find the optimum hyper plane, $\|w\|^2$ must be minimized under the constraint $y_i(w \cdot x_i + b) \geq 1$, $i = 1, 2, \ldots, n$. Therefore, it is required to solve the optimization problem given by

$$\begin{aligned} &\min \quad \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{5}$$

Now, the positive slack variables ξ_i are introduced to substitute in the optimization problem and allow the method to extend for a nonlinear decision surface. The new optimization problem is given as

$$\begin{aligned} &\min_{w,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0,\ i = 1, 2, \ldots, n, \end{aligned} \tag{6}$$

where C is a penalty parameter which manages the tradeoff between margin maximization and error minimization. Thus, the classification decision function becomes

$$f(x) = \operatorname{sign}\left(\sum_{i=1}^{N} L_i y_i K\left(x_i, x_j\right) + b\right), \tag{7}$$

where L_i are Lagrange multipliers, and $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ is a kernel function which can map the data into a higher dimensional space through some nonlinear mapping function $\varphi(x)$ for a nonlinear decision system. In present work, we use radial basis function (RBF) kernel function. Consider two samples $x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T$ and $x_j = [x_{j1}, x_{j2}, \ldots, x_{jd}]^T$. The RBF kernel is calculated using $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$, where $\gamma > 0$ is the width of Gaussian.

2.3. Weighted Support Vector Machines (WSVM). Traditional SVMs assume that each gene of a sample contributes equally to the tumor classification results. However, this is not desirable since the quality of genes has a significant impact on the performance of a learning algorithm, and the quality of different genes is not the same. In this work, we propose a novel WSVM based on PSO. Section 2.5 describes this process in more details give the training set $\{x_i, y_i\}_{i=1}^{N}$ and the weighted vector $\alpha \in R^d$ which fulfills $\sum_{i=1}^{d} \alpha_i = 1$ for $\alpha_i \geq 0$. With respect to (5), this optimization problem can be written as follows:

$$\begin{aligned} &\min \quad \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \quad y_i(w.\operatorname{diag}(\alpha) \cdot x_i + b) \geq 1, \quad i = 1, 2, \ldots, n, \end{aligned} \tag{8}$$

where $\operatorname{diag}(\alpha) = \begin{bmatrix} \alpha_1 & 0 & \ldots & 0 \\ 0 & \alpha_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \alpha_d \end{bmatrix}$.

Substituting (8) into (6) yields the following new optimization problem

$$\begin{aligned} &\min_{w,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{s.t.} \quad y_i(w \cdot \operatorname{diag}(\alpha) \cdot x_i + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0,\ i = 1, 2, \ldots, n \\ &\sum_{i=1}^{d}\alpha_i = 1,\ \alpha_i \geq 0. \end{aligned} \tag{9}$$

1	2	3	4	⋯	$n+2$
C	γ	a_1	a_2	⋯	a_n

Figure 1: Solution representation.

Finally, the classification decision function becomes

$$f(x) = \operatorname{sign}\left(\sum_{i=1}^{N} L_i y_i K'\left(x_i, x_j\right) + b\right), \tag{10}$$

where $K'(x_i, x_j) = \exp(-\gamma\sqrt{\sum_{k=1}^{d} a_k(x_{ik} - x_{jk})^2})$ is the weighted RBF kernel.

2.4. Particle Swarm Optimization (PSO). PSO, proposed by Kennedy and Eberhart [12], is inspired by social behavior among individuals like the birds flocking or the fish grouping. PSO consists of a swarm of particles that search for the best position according to its best solution. During each iteration, every particle moves in the direction of its best personal and global position. The moving process of a particle is described as

$$\begin{aligned} v_{id}^{t+1} &= w * v_{id}^{t} + C_1 \cdot \text{rand} \cdot \left(p_{id}^{k} - x_{id}^{t}\right) \\ &\quad + C_2 \cdot \text{rand} \cdot \left(p_{\text{gbest}}^{t} - x_{id}^{t}\right), \\ X_{id}^{t+1} &= \alpha V_{id}^{t} + x_{id}^{t}, \end{aligned} \tag{11}$$

where t denotes the tth iteration; C_1 and C_2 are learning factors; rand is positive random number between 0 and 1 under normal distribution. α is the constraint factor which can control the velocity weight. w denotes the inertial weight coefficient; x_{id} denotes the velocity of a particle i; v_{id} denotes the velocity of a particle i; p_{id} is the personal best position of particle i; $p_{g\text{best}}$ denotes the best one of all personal best positions of all particles within the swarm [19, 20].

2.5. Proposed Method. In this section, we introduce the proposed mRMR-PSO-WSVM method. The aim of this system is to optimize the SVM classifier accuracy by automatically (1) preselecting the number of genes using mRMR method, (2) estimating the best gene weights and optimal values for C and γ by PSO. First, the original microarray dataset is preprocessed by the mRMR filter. Each gene is evaluated and sorted according to mentioned mRMR criterions in Section 3, and the first fifty top-ranked genes are selected to form a new subset. In fact, mRMR is applied to filter out many unimportant genes and reduces the computational load for SVM classifier. Then, a PSO-based approach is developed for determination of kernel parameters and genes weight. Gene weighting is introduced to approximate the optimal degree of influence of individual gene using the training set. Without gene weighting, two decision variables C and γ are required to be optimized. If n genes are required to decide for gene weighting, then $n + 2$ decision variables must be adopted (see Figure 1). The value of n variables ranges between 0 and 1, where sum of them is equal to 1.

The range of parameter C is between 0.01 and 5,000, while the range of γ is between 0.0001 and 32. Figure 2 illustrates the solution representation. We used this representation for particles and allowed PSO to find right value for each variable.

We also define a threshold function $U_\delta(\cdot)$ to avoid using noisy genes with lower predictive power and to put more importance on the genes with higher discriminative power. In fact, $U_\delta(\cdot)$ works as gene selector which omits the redundant genes in the final step again. The domain of this function is the set of gene weights and the range is a revised weight for each gene

$$U_\delta(a_i) = \begin{cases} 0 & \text{if } a_i \le \delta, \\ a_i & \text{if } a_i > \delta, \end{cases} \tag{12}$$

where $0 \le \delta \le 1$ and a_i is the degree of importance of ith gene. Finally, the weighted vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ is determined by normal form as

$$\alpha_k = \frac{U_\delta(a_k)}{\sum_{i=1}^{d} U_\delta(a_i)}, \quad k = 1, 2, \ldots, d. \tag{13}$$

Therefore, as mentioned in Figure 2 the training process can be represented as follows

(1) Use the mRMR method to preselect fifty top-ranked genes. These selected genes then utilized in next stages where the PSO was employed to obtain optimal gene weights and kernel parameters.

(2) Involve the cross-validation method to separate dataset into training and testing set.

(3) Then, for each training set set up parameters of PSO. Generate randomly all particles' positions and velocity and set up the learning parameters, the inertia weight and the maximum number of iterations.

(4) Train WSVM classifier according to particles values.

(5) Calculate the corresponding fitness function formulated by ($classified/total$) ($total$ denotes the number of training samples, and $classified$ denotes the number of correct classified samples) for each particle.

(6) Update the velocity and position of each particle using (11).

(7) If the specified number of generations is not yet satisfied, produce a new population of particles and return to step (4).

(8) Select the gene weights and kernel parameters values from the best global position $p_{g\text{best}}$ and discard redundant genes with threshold function $U_\delta(\cdot)$.

(9) Train WSVM classifier with obtained parameters.

(10) Classify patients with the optimal model.

3. Experimental Results

The proposed mRMR-PSO-WSVM was implemented using the MATLAB software package version 7.2. We compared our suggested method with SVM, mRMR-SVM, mRMR-PSO-SVM classifiers to consider the effect of each component on classification results. We also extend our experiments by employing the classifiers that have been suggested before by Shen et al, [8] and Abdi and Giveki [18] which were denoted by PSO-SVM[1] and PSO-SVM[2] in Table 3, respectively. The discrete PSO was applied to select genes in PSO-SVM[1]. Each particle was encoded to a string of binary bits associated with the number of genes, which is made up of an SVM classifier with all its features. A bit "0" in a particle represented the uselessness of corresponding gene. Also, in PSO-SVM[2] Abdi and Giveki utilized PSO to determine SVM kernel parameters based on the fact that kernel parameters setting in training procedure significantly influence the classification accuracy [18].

Table 1: Detailed information of gene expression datasets.

Dataset name		Number of		
		Samples	Categories	Genes
Leukemia	Acute myeloid leukemia	25	2	7129
	Acute lymphoblastic leukemia	47		
Colon	Cancerous colon tissues	40	2	2000
	Normal colon tissues	22		

Table 2: PSO parameters.

Parameters	Values
Swarm size	50
The inertia weight	0.9
Accelration constants C_1 and C_2	2
Maximum number of iterations	70

The classifiers are evaluated on two popular public datasets: leukemia [21] and colon [22] datasets both of which consist of a matrix of gene expression vectors obtained from DNA microarrays for a number of patients. The first set was obtained from cancer patients with two different types of leukemia, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The complete dataset contains 25 AML and 47 ALL samples. The second set was obtained from cancerous and normal colon tissues. Among them, 40 samples are from tumors, and 22 samples are from healthy parts of the colons of the same patients [23]. The detailed information of them is collected in Table 1.

To calculate the accuracy of classifiers, the leave-one-out cross-validation (LOOCV) was involved using a single observation from the original sample as the testing data, and the remaining observations as the training data. This was repeated such that each observation in the sample was used once as the testing data. Moreover, in order to make experiments more realistic, we conducted each experiment 10 times on each dataset, and the average of classification accuracies of ten independent runs besides the average of number of selected genes as considered to evaluate the performance of classifiers. The related parameters of PSO algorithm applied in the experiments are also shown in Table 2.

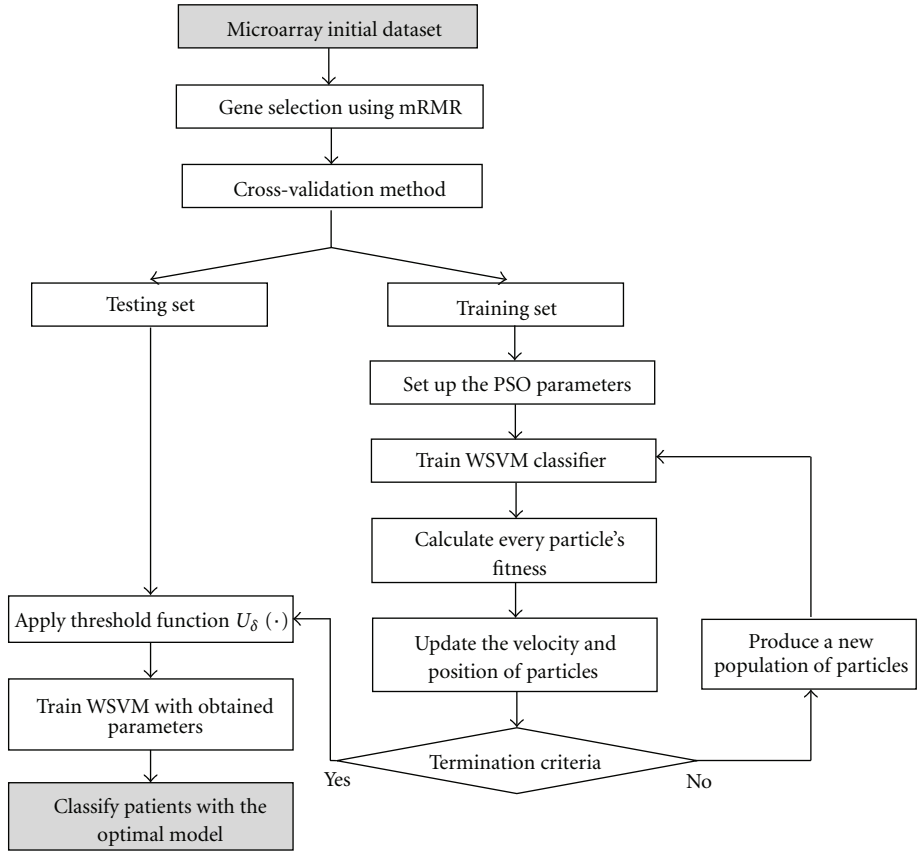


Figure 2: The process of classification by mRMR-PSO-WSVM.

Table 3: The values of the statistical parameters of the classifiers.

Methods/datasets	Leukemia		Colon	
	Acc (%)	Selected genes	Acc (%)	Selected genes
SVM	90.28	7129	83.87	2000
mRMR-SVM	97.22	50	83.87	50
PSO-SVM[1]	94.44	22.5	85.48	20.1
PSO-SVM[2]	93.06	7129	87.01	2000
mRMR-PSO-SVM	100	17.7	90.32	10.3
mRMR-PSO-WSVM	*100*	*3.8*	*93.55*	*6.2*

In addition, we filtered out all those genes having the PSO weight equal to or less than a quality threshold δ in the proposed method. To find the best value for this, we started from 0.2 and kept increasing this threshold value by 0.1 and saved the classification results. We found that for leukemia and colon datasets $0.3 \leq \delta \leq 0.5$ is always the best choice. Table 3 shows the classification accuracy of classifiers. As it can be observed, the classification accuracy of SVM on two datasets is not very interesting. Furthermore, the accuracy when the mRMR filter is employed generally outperforms the accuracy without gene selection. This implies that gene selection is able to improve the classification accuracy and mRMR is an effective tool to omit the redundant and noisy genes. In addition, the accuracy of PSO-SVM[1] shows that the selection of genes that are really indicative for tumor classification is a key step in developing a successful gene expression-based data and PSO is a promising tool for handling this. Also, the result of PSO-SVM[2] emphasizes on the fact that kernel parameters setting significantly influences the classification accuracy of SVM. Classification accuracy of the mRMR-PSO-SVM explains well the benefits of both gene selection and kernel parameters determination using PSO. In final, the proposed mRMR-PSO-WSVM achieves the highest classification accuracy together with lowest average of selected genes on test sets. This confirms that the suggested PSO-based gene weighting achieves better performance compared to binary PSO. Also, the average of selected genes shows that using the threshold function $U_\delta(\cdot)$ is very effective to reduce the number of selected genes.

Table 4: Classification accuracy of our method with other methods from literature (under 10-fold cross validation).

(Authors, year)	Method	Leukemia		Colon	
		Acc (%)	S. G.	Acc (%)	S. G.
(Ruiz et al., 2006) [24]	NB-FCBF	95.9	48.5	77.6	14.6
(Shen et al., 2007) [8]	PSOSVM	N. C.	N. C.	91.67	4.00
(Li et al., 2008) [15]	Single PSO	94.6	22.3	87.1	19.8
(Li et al., 2008) [15]	Single GA	94.6	23.1	87.1	17.5
(Li et al., 2008) [15]	Hybrid PSO/GA	97.2	18.7	91.90	18.00
(Shen et al., 2008) [17]	HPSOTS	98.61	7.00	93.32	8.00
(Abdi et al., 2012) [13]	*mRMR-PSO-WSVM*	*98.74* [18]	*4.1*	*93.55*	*6.8*

* S. G. and N. C. denote selected genes and not considered, respectively.

Table 5: Classification accuracy of our method with other methods from literature (under LOOCV).

(Authors, year)	Method	Leukemia		Colon	
		Acc (%)	S. G.	Acc (%)	S. G.
(Mohamad et al., 2007) [25]	IG + NewGASVM	94.71	20.00	N. C.	N. C.
(El Akadi et al., 2011) [17]	mRMR-GA	100	15.00	85.48	15.00
(Abdi et al., 2012) [18]	*mRMR-PSO-WSVM*	*100*	*3.8*	*93.55*	*6.2*

* S. G. and N. C. denote selected genes and not considered, respectively.

Tables 4 and 5 present the results of previously suggested methods besides the proposed mRMR-PSO-WSVM classifier. In order to make a more reliable comparison we try to carry out experiments with two cross-validation methods since some previously reported results were obtained under 10-fold cross validation and the other under LOOCV. Tables 4 and 5 show the results under 10-fold and LOO, respectively. We can see that the proposed classifier can obtain far better classification accuracy than previously suggested methods under both the cross-validation methods. Therefore, we can conclude that our method obtains promising results for gene selection and tumor classification problems.

4. Conclusion and Future Researches

This work presented a PSO-based approach to construct an accurate SVM in classification problems dealing with high-deminsional datasets especially gene expressions. This novel approach was a two-stage method in which, first of all, the mRMR filter technique was applied to preselect an effective genesubset from the candidate set. Then it formed a novel SVM in which PSO not only discarded redundant genes, but also especially took into account the degree of importance of each gene and assigned diverse weights to the different genes. It also used PSO to find appropriate kernel parameters since the choice of gene weights influences the optimal kernel parameters and vice versa. The experiments conducted using two different datasets for cancer classification show that the proposed mRMR-PSO-SVM outperforms the previously reported results. Experimental results obtained from UCI datasets or other public datasets and real-world problems can be tested in the future to verify and extend this approach.

References

[1] D. Hwang, W. A. Schmitt, G. Stephanopoulos, and G. Stephanopoulos, "Determination of minimum sample size and discriminatory expression patterns in microarray data," *Bioinformatics*, vol. 18, no. 9, pp. 1184–1193, 2002.

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

[3] M. P. S. Brown, W. N. Grundy, D. Lin et al., "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 1, pp. 262–267, 2000.

[4] V. Vapnik, *The Nature of Statistical Learning Theory*, New York, NY, USA, 1995.

[5] E. Byvatov, U. Fechner, J. Sadowski, and G. Schneider, "Comparison of support vector machine and artificial neural network systems for drug/nondrug classification," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1882–1889, 2003.

[6] C. Z. Cai, W. L. Wang, L. Z. Sun, and Y. Z. Chen, "Protein function classification via support vector machine approach," *Mathematical Biosciences*, vol. 185, no. 2, pp. 111–122, 2003.

[7] H. X. Liu, R. S. Zhang, F. Luan et al., "Diagnosing breast cancer based on support vector machines," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 3, pp. 900–907, 2003.

[8] Q. Shen, W.-M. Shi, W. Kong, and B.-X. Ye, "A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification," *Talanta*, vol. 71, no. 4, pp. 1679–1683, 2007.

[9] B. Jin and Y. Q. Zhang, "Support vector machines with evolutionary feature weights optimization for biomedical data classification," in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS '05)*, pp. 177–180, June 2005.

[10] H. J. Xing, M. H. Ha, D. Z. Tian, and B. G. Hu, "A novel support vector machine with its features weighted by mutual information," in *Proceedings of IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (IJCNN '08)*, pp. 315–320, June 2008.

[11] T. Wang, "Improving SVM classification by feature weight learning," in *Proceedings of the International Conference on Intelligent Computation Technology and Automation (ICICTA '10)*, pp. 518–521, May 2010.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 1944, pp. 1942–1948, December 1995.

[13] Q. Shen, W.-M. Shi, and W. Kong, "Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 53–60, 2008.

[14] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.

[15] S. Li, X. Wu, and M. Tan, "Gene selection using hybrid particle swarm optimization and genetic algorithm," *Soft Computing*, vol. 12, no. 11, pp. 1039–1048, 2008.

[16] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[17] A. El Akadi, A. Amine, A. El Ouardighi, and D. Aboutajdine, "A two-stage gene selection scheme utilizing MRMR filter and GA wrapper," *Knowledge and Information Systems*, vol. 26, no. 3, pp. 487–500, 2011.

[18] M. J. Abdi and D. Giveki, "Automatic detection of erythemato-squamous diseases using PSO–SVM based on association rules," *Engineering Applications of Artificial Intelligence*. http://www.sciencedirect.com/science/article/pii/S0952197612000218.

[19] J. Wei, Z. Jian-Qi, and Z. Xiang, "Face recognition method based on support vector machine and particle swarm optimization," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4390–4393, 2011.

[20] M. J. Abdi and H. Salimi, "Farsi handwriting recognition with mixture of RBF experts based on particle swarm optimization," *International Journal of Information Science and Computer Mathematics*, vol. 2, no. 2, pp. 129–136, 2010.

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

[22] U. Alon, N. Barka, D. A. Notterman et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 12, pp. 6745–6750, 1999.

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

[24] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Incremental wrapper-based gene selection from microarray data for cancer classification," *Pattern Recognition*, vol. 39, no. 12, pp. 2383–2392, 2006.

[25] M. S. Mohamad, S. Omatu, S. Deris, and S. Z. M. Hashim, "A model for gene selection and classification of gene expression data," *Artificial Life and Robotics*, vol. 11, no. 2, pp. 219–222, 2007.