Improving Prediction of Drug Therapy Outcome via Integration of Time Series Gene Expression and Protein Protein Interaction Network

Liwei Qian[1], Haoran Zheng[1,2,*]
[1]School of Computer Science and Technology and [2]Department of Systems Biology,
University of Science and Technology of China, Hefei 230026, PR China.
[*] Corresponding author. E-mail: hrzheng@ustc.edu.cn

Abstract—**Drug therapy to patients is often with partial success, and has been associated with a number of adverse reactions. Prediction of patients' response to therapy at the early stage of the treatment is crucial to avoiding those unnecessary adverse reactions. In this paper, a new approach that integrates time series gene expression and Protein Protein Interaction (PPI) network is presented to improve the prediction of patients' response to drug therapy. Experimental results showed that our method outperformed previous approaches. The method proposed here offers a huge potential for applications in pharmacogenomics and medicine.**

***Keywords*—clinical studies, time series, gene expression, PPI network.**

I. Introduction

The use of gene expression profiling allows clinical diagnosis to be made on a molecular level, thereby facilitating choice of treatment based on the patients' genetic traits [1]. Gene expression experiments were, until more recently, limited to static analysis [2]. Gene markers selected by these approaches may be functionally related, hence contain redundant information, leading to the degradation of the overall classification performance [3]. Due to this type of difficulty, many research groups [4, 5] proposed to combine gene expression measurements with biological networks, such as protein-protein interaction network, metabolic network and achieved good results. Nonetheless, those integrated approach mainly focus on static gene expression analysis. While those static analysis are appropriate for some cases, they are less appropriate for longer term clinical follow-up [6]. The last few years have witnessed an increase in time series gene expression experiments and analysis. In fact, taking the temporal dynamics of gene expression into account enables the study of complex biomedical problems, such as drug response, from a new and different perspective. Hence, a new approach of classification of time series data that integrates biological network needs to be proposed.

Here, we investigate the problem of classification of Multiple Sclerosis (MS) patients with respect to their response to interferon beta (IFN-β) treatment. MS is one of the most prevalent autoimmune disorders. In spite of its chronic clinical course, drug therapy with IFN-β is widely applied to reduce the intensity and frequency of exacerbations. Nevertheless, almost one-half of the patients do not respond to the therapy [7]. Furthermore, the drug therapy has been associated with a number of adverse reactions, including flu-like symptoms, transient laboratory abnormalities, menstrual disorders, increased spasticity, and dermal reactions [8]. Hence, prediction of patients' response to therapy ahead of the treatment or at the early stage of the treatment is crucial to avoiding unnecessary adverse reactions.

Based on the above consideration, we propose a new approach (Fig. 1) to predict the response to drug therapy. Firstly, a Hidden Markov Model/Gaussian Mixture Model (HMM/GMM) hybrid model is applied to transforming gene expression data into relative expression level. In addition, we put forward a new time and space efficient biclustering method to extract bicluster from the relative expression level. Furthermore, a novel SVM/KNN hybrid classifier, which taking PPI network into consideration, is trained to predict the patients' response to drug therapy. Experimental results demonstrate that the proposed integrated framework, achieved large improvements over previous methods.

Figure1. Schematic overview of prediction of patients' response to therapy.

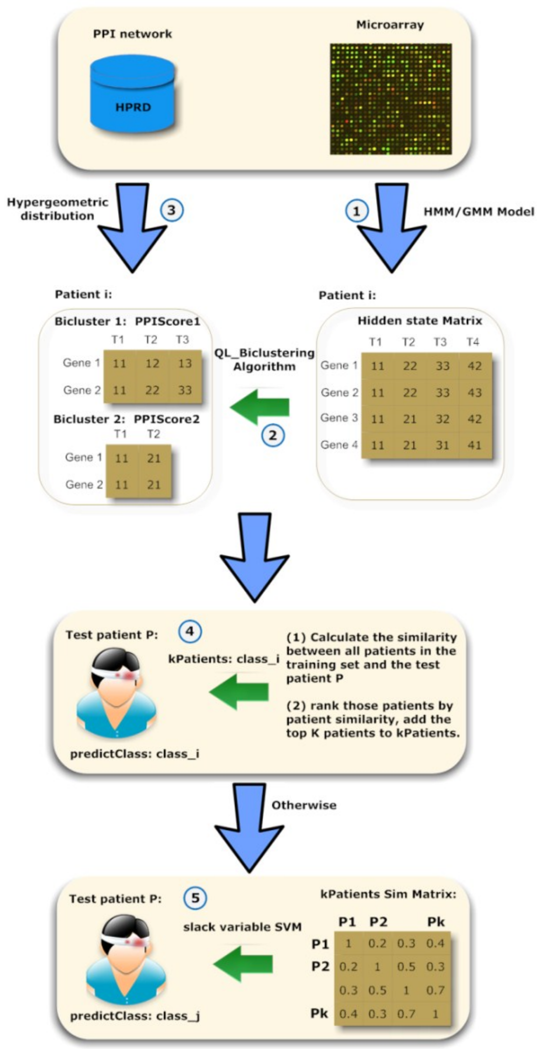
II. Methods

A. *Inference of gene expression levels via Hidden Markov Model/Gaussian Mixture Model (HMM/GMM) hybrid model*

The HMM/GMM hybrid model is a classical method in speech recognition [9]. Here, the model is introduced to process time series gene expression data, in detail, discretizing the contiguous gene expression values and exploring time-dependence property of the data.

978-1-4673-4398-5/12/$31.00 ©2012 IEEE

A standard continuous HMM is characterized by the following elements [10]: 1) Q, the number of hidden states. 2) A=[aij], the state transition probability distribution, where aij is the transition probability from state i to j. 3) B={bj(x)}, the emission probability distribution, where bj(x) is the emission probability to observe x in state j. 4) $\pi = \{\pi_i\}$, the initial state distribution, where πi is the start point probability of the state i. The HMM/GMM hybrid model [9] is a specific form of HMM model, which the emission probability distribution to observe x in state j can be modeled by a mixture of Gaussian functions, $b_j(x) = \sum_{i=1}^{n} w_{ji}\, g_{ji}\left(\mu_{ji}, \sigma_{ji}\right)$, where n is the mixture number, wji is the mixture weight, gji is the component Gaussian function with expected value μji and variance σji.



We assume that expression values of each gene are from a Markov process. Each gene stays in one state at a time point. The gene can stay in the same state or switch to other states in the next time point (the number of states is Q). The model is initialized as follows: the initial state distribution π, state transition probability distribution A and mixture weight distribution w are set to uniform distribution. The expression values of all genes are divided into Q bins. Each bin has the same number of figures. The mixture number n is equal to the number of genes. μji, σji are set to the mean and variance of expression values in the j-th bin of gene i, respectively. The model parameter is firstly trained by EM algorithm [11]. Then Viterbi algorithm [11] is applied to infer the hidden state sequences for all genes. In our later experiments, we set Q=2.

B. *Extraction of local gene expression patterns through a new Biclustering algorithm*

A new biclustering algorithm named QL_Biclustering algorithm is proposed to extract biclusters from the hidden state sequences of all genes, which is obtained at previous step. A bicluster B(G, J(T, S)) consists of gene set G, consecutive time point set T and gene state sequence S, which is made up of the state values of a gene at the time points of T, where all genes of G share the same S. In order to differentiate time points, a transformation is introduced that appends time point to each gene state. For example, the state sequence at first three time points of a certain gene may be S= {3, 2, 1}. The transformed state sequence of that gene is J (T, S) = {13, 22, 31}.

ALGORITHM I: QL_BICLUSTERING ALGORITHM

```
1.  for each state sequence do
2.    Construct all responding suffix strings (SA)
3.  end for
4.  Sort all the suffix strings by MSD (most significant digit) radix sort
    method
5.  for all neighbor suffix strings do
6.    Compute longest common prefix and store the longest common
    prefix length (LCPLength)
7.  end for
8.  for each distinct values i in LCPLength do
9.    for each occurences (j) of i in LCPLength do
10.     pos(i,j) = pinpoint(LCP,i,j)
11.     if i < ml then {Blcp[pos(i,j)] = 1;   continue; }
12.     l = max{k| k < pos(i,j)   and   Blcp[k] = 1} + 1
13.     r = min{k| k > pos(i,j)   and   Blcp[k]=1}
14.     Blcp[pos(i,j)]= 1
15.     if ( r – l + 1 ) < mo then continue;
16.     if   l > 1 && LCPLength( l -1 ) == i then continue;
17.     Bicluster B(G,J):
18.       Gene set(G): Gene[ l … r];
19.       J(Timepoint(T) and corresponding state value(S) ): SA[ pos(i,j) ,
    1…i]
20.   end for
21. end for
```

QL_Biclustering Algorithm (Algorithm I) is based on the suffix string and longest common prefix, the input of which is gene state sequences, ml (minimum number of consecutive time points), mo (minimum number of genes), the output of which is all biclusters satisfying user’s requirements. We traverse all the values in LCPLength from the smallest to the largest. For each occurrence of each different value in LCPLength, we process it as follows: At step 10, the jth occurrence position of value i in LCPLength is pinpointed and stored in pos(i, j). The number of time points of current bicluster is ensured to be not less than ml at step 11. We find an interval [l, r] at steps 12-13, the suffix strings among which share a common prefix with length i. Any bicluster,

978-1-4673-4398-5/12/$31.00 ©2012 IEEE

gene number of which is less than mo, is ignored at step 15. Step 16 verifies whether the candidate bicluster can be extended to the left. If it does, the candidate bicluster will be discarded, for which it is not maximal.

QL_Biclustering algorithm performs well both in time and space aspects. In our experiments, both the parameter ml and mo are set to 2.

C. Integration of biological network

Integration of a biological network leads to good classification performance and improves biological interpretability of the results. In this article, we merely use PPI network as an illustration, but the proposed method is independent of the nature of the network and can be extended to combine many other biological networks.

Among all biclusters extracted at previous step, this kind of biclusters, within which genes are highly connected relative to overall connectivity in the PPI network, is more preferable. In consideration of this, a modified hypergeometric distribution is adopted to model the interaction between gene i and bicluster B.

$$p(i\text{->}B) = \frac{\binom{n_{i\text{->}B} + n_{B\text{->}B}}{n_{i\text{->}B}}\binom{n_{i\text{->}B'} + n_{B\text{->}B'}}{n_{i\text{->}B'}}}{\binom{n_{i\text{->}B} + n_{B\text{->}B} + n_{i\text{->}B'} + n_{B\text{->}B'}}{n_{i\text{->}B} + n_{i\text{->}B'}}} \tag{1}$$

particularly, $p(i\text{->}B)=0$, when $n_{i\text{->}B}=0$

where B' is the gene set consisting of all genes except genes in bicluster B, $n_{i\to B}$ is the number of interactions between gene i and genes in bicluster B except gene i in the network, the rest may be deduced by analogy.

For each bicluster B, its preference (PPIScore) can be scored as follows:

$$\text{PPIScore}(B) = \sum_{i \in B} p(i \to B)/|B| \tag{2}$$

In terms of a specific bicluster, the higher the PPIScore is, the more closely connected genes in the bicluster are, the more likely the genes share a common biological function.

D. Classification algorithm

1) Computation of similarity between two patients: Jaccard Index [12] is used to compute the similarity measure $\text{Sim}(B_i,B_j)$ between two biclusters $B_i(G_i,J_i)$ and $B_j(G_j,J_j)$. The PPIScore of all biclusters of patients P_1 and P_2 should be normalized. In such circumstances, similarity between two patients P_1 and P_2 that integrates PPI network can be calculated as follows:

$$\begin{aligned}&\text{PPISim}(P_1,P_2)=\\&\frac{1}{2}\Big(\sum_{i=1}^{n}\text{PPIScore}(B_{1i})*\max(\text{Sim}(B_{1i},B_{2j}),j\in 1,\ldots,m)\\&+\sum_{j=1}^{m}\text{PPIScore}(B_{2j})*\max(\text{Sim}(B_{2j},B_{1i}),i\in 1,\ldots,n)\Big)\end{aligned} \tag{3}$$

n, m is the number of biclusters in P_1, P_2 respectively. B_{1i}, B_{2j} denotes the ith bicluster in P_1, the jth bicluster in P_2 respectively.

2) PPI-SVM-KNN classifier: A hybrid of SVM and KNN classifier integrating PPI network named PPI-SVM-KNN is used here (Algorithm II). The KNN classifier is used at an initial stage, and then an SVM classifier is trained on the collection of the nearest neighbors. In the neighborhood of a small number of samples, SVM, more often than not, performs better than other classification methods [13]. This is a process of a coarse and quick categorization followed by a finer but slower classification, which inherits the advantages of both SVM and KNN classifiers. Specifically, linear SVM with slack variable is utilized here; PPISim among different patients is used as the kernel matrix. Since experimental results showed that nonlinear SVMs, such as the SVM with Gaussian kernel, do not lead to better performance, we use the simplest linear kernel. The model of linear SVM with slack variable:

$$\text{maximize} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \text{Kernel}(x_i,x_j) \tag{4}$$

$$\text{such that}: \sum_{i=1}^{n}\alpha_i y_i = 0,\ 0 \le \alpha_i \le C.$$

where $\text{Kernel}(x_i,x_j)=\text{PPISim}(x_i,x_j)$,

X are training samples, y are corresponding labels.

The label of a new sample P can be predicted as follows:

$$\text{predictClass} = \text{sign}\Big(\sum_{i\in SV}\alpha_i y_i \text{Kernel}(x_i,P)+b\Big) \tag{5}$$

where

$SV(\text{Support Vector})=\{i|\alpha_i \neq 0\}$.

$b = y_i - \sum_{j=1}^{n}\alpha_j y_j \text{Kernel}(x_i,x_j)$,

for any i that $\alpha_i \neq 0$.

The parameter C can be viewed as a way to control "softness": it trades off between the relative importance of maximizing the margin and fitting the training data (minimizing the error). The parameter K specifies the classifier's dependence on choice of the number of neighbors. When K is small, the algorithm behaves like a straightforward KNN classifier. When K is large enough, our method is totally a SVM. From now on, we fix K=6 and C=100 for any experiment performed later.

ALGORITHM II: PPI-SVM-KNN

```
Input: C, K ; Output: predictClass;
for each test patient P do
  Calculate the similarity between all patients in the training set and the test patient P and rank those patients by similarity measure, add the top K patients to kPatients.
  if the label of all the patients in kPatients is class_i
      predictClass = class_i;
  else
      Compute similarity matrix PPISim among patients in kPatients;
      Train linear SVM with slack variable classifier using PPISim as kernel and predict the label of test patient P
  end if
end for
```

III. Results

A. Dataset:

We test our integrated framework on a set of microarray expression time series data from [14], predicting whether Multiple Sclerosis(MS) patients will respond positively to treatment with recombinant human interferon beta (rIFNβ). The dataset contains time expression profiles of 52 multiple sclerosis patients, out of which 33 are good and 19 are poor responders to rIFNβ. Expression profiles of 70 genes were measured for up to seven times per patient. The first five observations were at a regular interval of 3 months each, while the last 2 observations were spaced 6 months apart. 17 patients missed a test and hence have only 6 measurements, 8 patients missed two tests and hence have only 5 measurements. All missing values are filled with a weighted mean of the three closest neighboring values after data normalization.

Since our framework is tested on the expression profiles of MS patients, human protein-protein interaction network is chose and obtained from HPRD [15]. In addition, we exclude those protein-protein interactions, which are not related to the 70 genes of MS dataset, from the entire human protein interaction network.

B. Experimental results

A commonly used strategy to evaluate the classification performance is the k-fold cross validation (CV) scheme. In this work, we use 10 repetitions of stratified 4-fold CV, as performed in previous approaches.

Table I: Classification accuracy of distinct methods in the MS patient response data

Method	Accuracy	Author
IBIS	74.20	Baranzini
dsSVM	73.44(2.56)	Borgwardt
KNN SPTL	68.46(2.19)	Carreiro
Meta-Profiles	59.42(6.17)	Carreiro
HMMClass	78.42(3.44)	Lin
PPI-SVM-KNN	86.42 (2.06)	

In Table I, we demonstrated the classification accuracy of distinct methods in the MS patient response data. All those methods except IBIS [14] are based on all 70 genes without any feature selection. The IBIS method proposed by Baranzini et al. used only the first time point expression data and 3 selected genes. Lin et al. [6] based their classifier on hidden markov models using discriminative learning. Borgwardt et al. [16] introduced an approach of SVM, based on dynamic systems kernels. Carreiro et al. [17] proposed a biclustering-based classification algorithm. As shown in Table III, PPI-SVM-KNN outperformed other methods. Specifically, accuracy of PPI-SVM-KNN is 86.42, higher than other previous methods. Furthermore, our method is more stable than other classifiers, since the standard deviation of PPI-SVM-KNN is less than all other approaches.

As prediction accuracy is not enough to measure the quality of a classifier, some other performances of our framework were tested, Recall, Precision, F-measure, which are commonly used in evaluating the classification performance. As shown in Fig. 2, our framework performed well in those metrics. We emphasize that the Recall value is remarkably high, a truly good responder being predicted as bad responder seldom happens, which is beneficial to the timely treatment of patients.

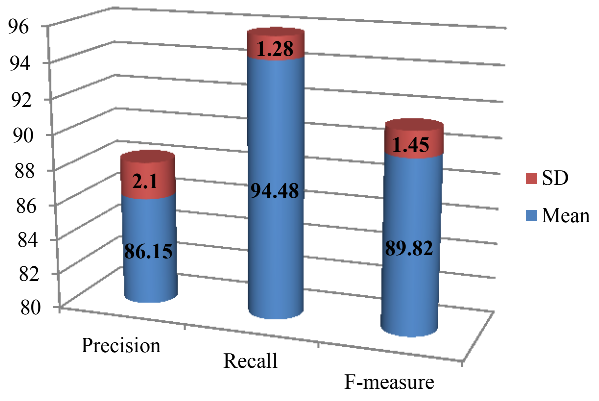
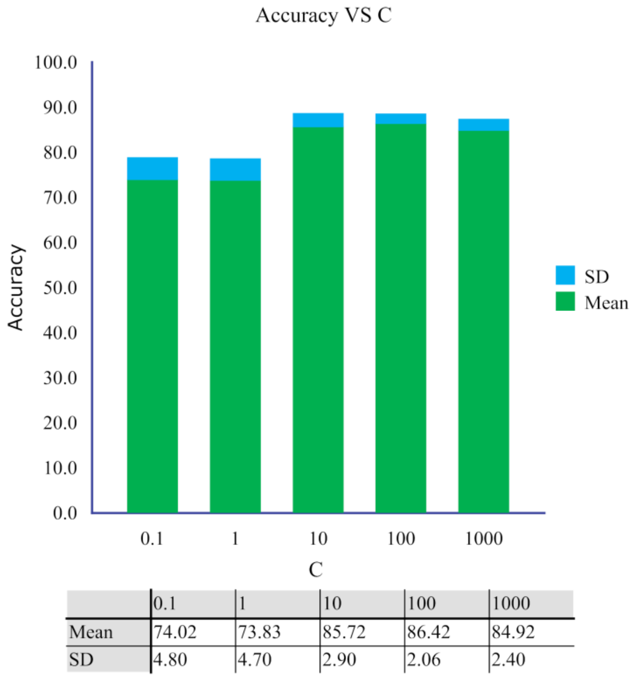


Figure 2: Precision, Recall, and F-measure of PPI-SVM-KNN

We also checked the influence of parameter C on classification accuracy. The parameter C of PPI-SVM-KNN trades off between the relative importance of maximizing the margin and fitting the training data (minimizing the error). As shown in Fig. 3, classification accuracy varied little and was relatively stable with changes in the value of C. Furthermore, as long as C is larger than 1, our framework outperformed other previous approaches.



	0.1	1	10	100	1000
Mean	74.02	73.83	85.72	86.42	84.92
SD	4.80	4.70	2.90	2.06	2.40

Figure 3: The influence of parameter C on classification accuracy.

Finally, we checked the contribution of integrating PPI network. Our integrated framework prefers those local expression patterns (biclusters), the genes of which tend to

978-1-4673-4398-5/12/$31.00 ©2012 IEEE

share a common biological function. Here we compared the results of integrating PPI network to that of without integrating PPI network. As shown in Fig. 4, the prediction accuracy is higher and slightly more stable when integrating PPI network with all other parts unchanged. Therefore, integrating PPI network into our framework of prediction indeed makes a difference.

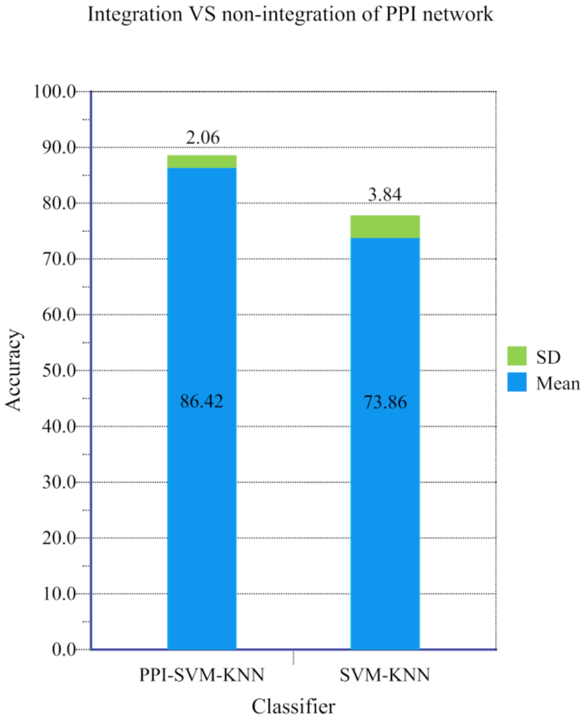


Figure 4: The contribution of integrating PPI network

IV. Conclusion

In this paper, a new approach that integrates time series gene expression and PPI network is presented to improve the prediction of patients' response to drug therapy. In comparison with previous approaches, our method performed better with regard to prediction accuracy and its stable performance. Besides, the proposed method with significantly high Recall value, a truly good responder being predicted as bad responder seldom happens, is beneficial to the timely treatment of patients. In addition, we demonstrate that integration of biological network in our method definitely makes a difference in drug response prediction. We would like to point out that the potential of our method is shown only in drug response prediction here, but it offers far more possibilities for applications in pharmacogenomics and medicine.

ACKNOWLEDGMENT

We gratefully acknowledge funding support from the National Key Technologies R&D Program (2011CB910200).

References

[1] R. Spang, "Diagnostic signatures from microarrays: a bioinformatics concept for personalized medicine," Biosilico, vol. 1, pp. 64-68, 2003.

[2] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, and X. Yu, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," Nature, vol. 403, pp. 503-511, 2000.

[3] S. Junjie, Y. Byung-Jun, and D. Edward, "Identification of diagnostic subnetwork markers for cancer in human protein-protein interaction network," BMC Bioinformatics, vol. 11.

[4] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker, "Network-based classification of breast cancer metastasis," Molecular systems biology, vol. 3, 2007.

[5] P. Dao, K. Wang, C. Collins, M. Ester, A. Lapuk, and S. C. Sahinalp, "Optimally discriminative subnetwork markers predict response to chemotherapy," Bioinformatics, vol. 27, pp. i205-i213, 2011.

[6] T. Lin, N. Kaminski, and Z. Bar-Joseph, "Alignment and classification of time series gene expression in clinical studies," Bioinformatics, vol. 24, pp. i147-i155, 2008.

[7] J. Río, C. Nos, M. Tintoré, C. Borrás, I. Galán, M. Comabella, and X. Montalban, "Assessment of different treatment failure criteria in a cohort of relapsing–remitting multiple sclerosis patients treated with interferon β: Implications for clinical trials," Annals of neurology, vol. 52, pp. 400-406, 2002.

[8] D. C. Mohr, W. Likosky, A. C. Boudewyn, P. Marietta, P. Dwyer, J. Van Der Wende, and D. E. Goodkin, "Side effect profile and adherence to in the treatment of Multiple Sclerosis with interferon beta-la," Multiple sclerosis, vol. 4, pp. 487-489, 1998.

[9] E. Rodríguez, B. Ruíz, Á. García-Crespo, and F. García, "Speech/speaker recognition using a HMM/GMM hybrid model," 1997, pp. 227-234.

[10] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, pp. 257-286, 1989.

[11] S. Brunak, Bioinformatics: the machine learning approach: MIT press, 2001.

[12] S. C. Madeira, M. C. Teixeira, I. Sa-Correia, and A. L. Oliveira, "Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm," Computational Biology and Bioinformatics, IEEE/ACM Transactions on, vol. 7, pp. 153-165, 2010.

[13] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," 2006, pp. 2126-2136.

[14] S. E. Baranzini, P. Mousavi, J. Rio, S. J. Caillier, A. Stillman, P. Villoslada, M. M. Wyatt, M. Comabella, L. D. Greller, and R. Somogyi, "Transcription-based prediction of response to IFNβ using supervised computational methods," PLoS Biology, vol. 3, p. e2, 2004.

[15] T. S. K. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, and A. Venugopal, "Human protein reference database—2009 update," Nucleic acids research, vol. 37, pp. D767-D772, 2009.

[16] K. Borgwardt, S. Vishwanathan, and H. P. Kriegel, "Class prediction from time series gene expression profiles using dynamical systems kernels," 2006.

[17] A. Carreiro, O. Anunciação, J. Carriço, and S. Madeira, "Prognostic Prediction through Biclustering-Based Classification of Clinical Gene Expression Time Series," Journal of integrative bioinformatics, vol. 8, p. 175, 2011.

978-1-4673-4398-5/12/$31.00 ©2012 IEEE