Academy of Scientific Research and Technology
27th National Radio Science Conference
Faculty of Electronic Engineering, Menoufia Univ., Menouf, Egypt
16-18 March 2010

A New Software to Construct Gene Regulatory Networks From Microarrays Data

Fadhl M. Al-Akwaa[1,2], Nahed H. Solouma[2], Yasser M. Kadah[2]

[1]Biomedical Eng. Dept., Univ. of Science & Technology, Sana'a, Yemen (E-mail f_alakwa@k-space.org)
[2] Biomedical Engineering Department, Cairo University, Giza, Egypt

Abstract

As basic building blocks of life, genes, as well as their products (proteins), do not work independently. Instead, in order for a cell to function properly, they interact with each other and form a complicated network. Recently, many researchers agree that most of biological ambiguous questions might be easily being answered if a sophisticated modeling of the gene regulatory network (GRN) was constructed. Also, GRN was used to help understanding disease ontology and reducing the cost of drug development. During the last decade, many GRN inference algorithms that are based on microarrays data have been developed to unravel the complexity of gene regulation. One of the major problems with microarrays is that a dataset consists of relatively few time points with respect to a large number of genes. The dimensionality and high degree of noise are interesting problems in GRN modeling. In this paper we proposed a new integrated algorithm to overcome these problems. Our software show good performance comparable to previous methods and many of produced edges have evidence in the literature data. The proposed method was applied to time series gene expression data of *Saccharomyces Cerevisiae* and could potentially be applied to other networks in yeast as well as higher organisms.

1. Introduction

One of the main research areas of bioinformatics is functional genomics; which focuses on the interactions and functions of each gene and its products (mRNA, protein) through the whole genome (the entire genetics sequences encoded in the DNA and responsible for the hereditary information). In order to identify the functions of certain gene, we should be able to capture the gene expressions which describe how the genetic information converted to a functional gene product through the transcription and translation processes. Functional genomics uses microarrays technology to measure the genes expressions levels under certain conditions and environmental limitations. In the last few years, Microarray has become a central tool in system biology; consequently, the corresponding data analysis becomes one of the important work disciplines in bioinformatics.

The major goal of system biology is to understand how genes and its products interact to regulate cell cellular process. To achieve this goal it is necessary to reconstruct the gene regulatory networks (GRN) which help us to understand the working mechanisms of the cell in patho-physiological conditions. The structure of the GRN can be described as a wiring diagram that shows the direct and indirect influences on the expression of a gene and that describes which genes, in turn, can be regulated by such a gene through its translated protein or transcribed RNA product [1].

Within the last few years, a number of sophisticated approaches for the reverse engineering of cellular networks from gene expression data have been emerged. This may include Boolean networks[2], Bayesian networks[3], association networks[4], linear models[5], and differential equations[6].

A broad overview of the steps involved in the modeling of GRN is shown in Figure 1. The first step start with the question biologist required to be answered. For example, which genes are involved in controlling the cell cycle[7] and which genes are showed a similar drastic response to the environmental changes[8]. The next step is to prepare appropriate experiments relative to biological question. Next is the extraction of the gene expression matrix (rows are genes and columns are the different experiments or cells) from the microarray experiments using image processing techniques. Removing of non-informative genes and conditions, normalization and data denoising are described in the preprocessing step. The next step is the partitioning of the data set in to small overlapped biclusters.

In learning step the produced biclusters were learned using the reverse engineering approaches to produce the corresponding subnetworks i.e each bicluster produce one subnetwork. Learning step will be followed by integration of the generated subnetworks to produce the whole network. Assessment the performance of the resultant network using existing interactome databases and literature databases is performed in the validation step. Last step is that the generated validated network will open new hypotheses which need to be verified. For more details about these steps, the reader could referee to[9].
The reconstruction of gene networks is in general complicated by the high dimensionality of high-throughput data, i.e. a dataset consists of relatively few time points with respect to a large number of genes. Together with observational noise, these complications impose a limit on the reconstruction of gene network. In this study we propose a new algorithm to reduce data noise using spectral subtraction and data dimension using biclustering algorithms.

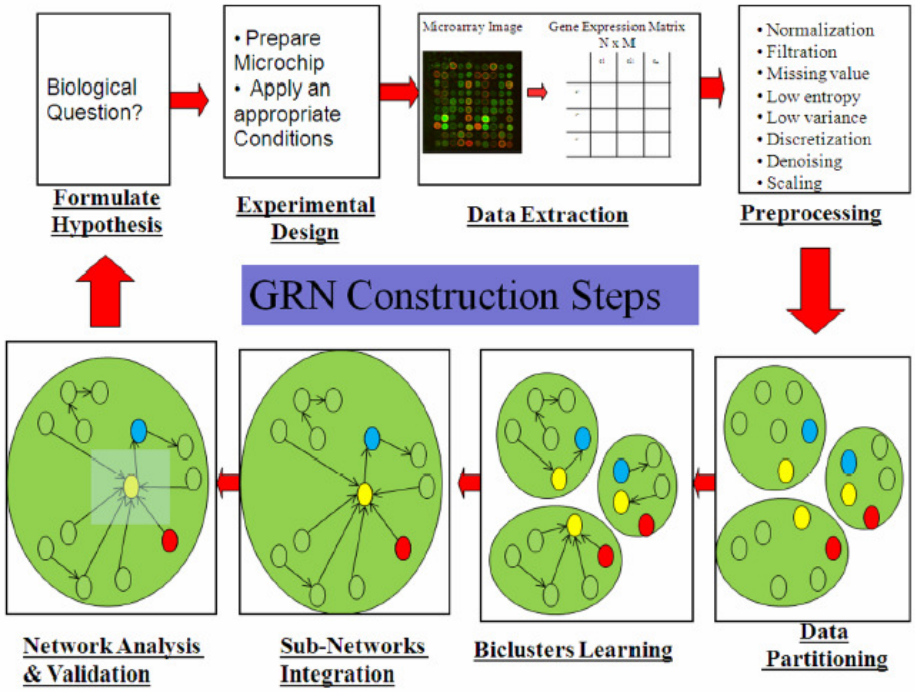

Fig. 1. Basic Steps involved in GRN construction

2. Methodology

2.1. Data Acquisition

The microarray data used in this work is a well-known dataset of yeast microarray gene expression provided by Spellman et al.[7], which can downloaded from Stanford Microarray Database http://smd.stanford.edu. The Spellman dataset consists of four synchronization experiments (alpha factor arrest, elutriation and arrest of CDC15 and CDC28 temperature-sensitive mutants) which were performed for a total of 73 microarrays during cell-cycle.

2.2. Data Prepossession

Measurements of microarrays may be biased by diverse effects such as efficiency of RNA extraction, reverse transcription, label incorporation, exposure, scanning, spot detection, etc. This necessitates the preprocessing of microarrays prior to data analysis. The datasets used in this work have been already preprocessed for background correction and normalization. In addition, further preprocessing steps should be applied for data refinement. In this paper, we applied the commonly used preprocessing such as gene filtration and missing value imputation[9, 10].

2.3. Data Denoising

For data cleaning, we developed a new denoising algorithm for further preprocessing and refinement. The proposed algorithm is an adaptive signal-preserving technique for gene expression data based on spectral subtraction. The new technique uses the uncorrelatedness of the random noise and the deterministic components of the signal to separate the two in the original power spectrum.

Figure 2 is a block diagram for the proposed denoising method. We will consider a model that is composed of the sum of one deterministic component ***d(t)*** incorporating both the true gene expression signal and the experimental noise and an uncorrelated stochastic component ***n(t)***. That is:

$$\mathbf{s(t) = d(t) + n(t)}$$

Since these two component are assumed to be independent, the corresponding power spectra are related by

$$P_{ss}(w)=P_{dd}(w)+P_{nn}(w)$$

where cross terms vanish because the two components are assumed uncorrelated. Hence, an estimate of the power spectrum of the deterministic component takes the below form[11]

$$P_{dd}(w)=P_{ss}(w)-P_{nn}(w)$$

That is, the signal power spectrum is obtained by spectral subtraction of the noisy signal and noise power spectra. In order to compute the deterministic signal component from its power spectrum, the magnitude of the Fourier transform can be obtained as the square root of the power spectrum. The problem now becomes that of reconstructing the signal using magnitude only information about its Fourier transform. Several techniques can be used to do that. The one used for this paper relies on an estimate obtained from the phase of the Fourier transform of the original signal ***S(w)*** . Hence, the Fourier transform of the processed signal $\boldsymbol{S_d(w)}$can be expressed as

$$S_d(w) = \sqrt{p_{dd}(w)}e^{j\,phase(s(w))}$$

The enhanced deterministic signal $\boldsymbol{s_d(t)}$ is then computed as the real part of the inverse Fourier transformation of this expression.

The following are the steps needed to implement the spectral subtraction denoising procedure in practice.

Step 1) Compute the variance of the gene expression values of each gene. Averaging the estimate from all time points to obtain the noise power spectrum level.

Step 2) Compute the Fourier transform of the gene time course and save the phase and magnitude parts of the result separately.

Step 3) Compute the original power spectrum of this time course using the periodogram method as the square of the magnitude of the Fourier transform in Step 2.

Step 4) Compute the denoised power spectrum by subtracting the noise power spectrum from Step 1. Observe any scaling factors that are introduced by the Fourier transform definition before performing the subtraction in addition to the factor α in (5) (use a default value of α =1).

Step 5) Compute the denoised signal Fourier transform as the square root of the denoised power spectrum from Step 4 multiplied by the phase retained in Step 2.

Step 6) Compute the denoised time course as the real part of the result of Step 5.

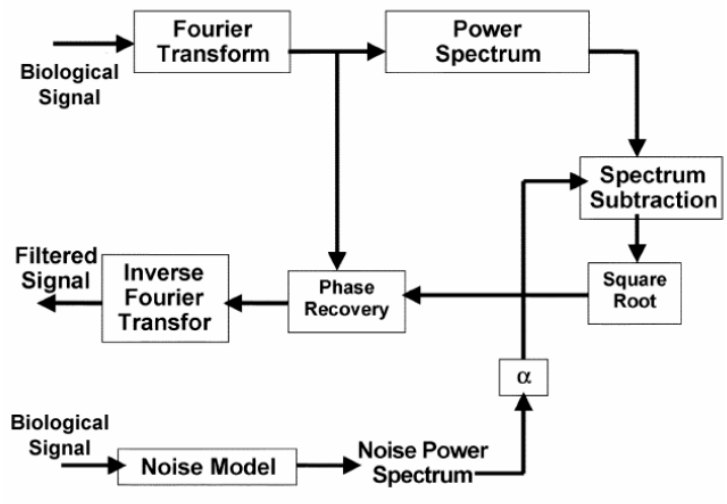

Fig. 2. Spectral Subtraction Denoising Algorithm Block Diagram

2.4. Data Partitioning

Clustering algorithms [12-14] were used to reduce data dimension. This is based on the assumption that genes which show similar expression patterns are co-regulated or part of the same regulatory pathway. But unfortunately, this is not always true. There are two limitations obstacle using clustering algorithms with microarrays data. First, all conditions are given equal weights in the computation of gene similarity; whereas, most conditions do not contribute information but instead increase the amount of background noise. Second, each gene is assigned to a single cluster, whereas in fact genes may participate in several functions and should thus be included in several clusters[15].

A new modified clustering concept to uncover processes that are active only over some but not all samples emerged which is called biclustering. A bicluster is defined as a subset of genes that exhibit compatible expression patterns over a subset of conditions[16].

During the last ten year, many biclustering algorithms have been proposed (see[17] for a survey), but the important questions are: which algorithm is better? And do some algorithms have advantages over others?

Generally, comparing different biclustering algorithms is not straightforward as they differ in strategies, approaches, time complexity, number of parameters and prediction ability. They are strongly influenced by user-selected parameter values. For these reasons, the quality of biclustering results is also often considered more important than the required computation time. Although there are some analytical comparative studies to evaluate the traditional clustering algorithms [18-20], for biclustering, no such extensive comparison exist even after initial trails have been taken. At the end, biological merit is the main criterion for evaluation and comparison between the various biclustering methods.

We have developed a comparative tool BicAT-Plus[21] that includes the biological comparative methodology and to be used as an extension to the BicAT program[22]. BicAT-Plus and its manual can be downloaded from the below two links:

http://home.k-space.org/BicAT-plus.zip and http://home.k-space.org/Bicat-plus-manual.pdf.

BicAT is a java biclustering toolbox which contains five biclustering (XMOTIF, ISA, OPSM, BIMAX, CC) and two traditional clustering algorithms (K-means and HCL).

For more details about BicAT-Plus the reader could refer our previous paper[21].

2.5. Network Generation

Many reverse engineering approaches to learn cellular networks from gene expression data have emerged.

Bayesian networks (BN) which were first used by Friedman et al~[3] has been widely used because of solid basics in statistics. Also BN able to handle missing data and permit the incomplete knowledge about the biological system. There are two important components to represent BN, the qualitative part which is called the directed acyclic graph (DAG) and the quantitative part which is the conditional probability of children given its parents. The popular approach to find the best DAG is to search the

DAG space and find the DAG which has the best score. Due to huge DAG space we have to use the heuristic searches. k2 algorithm, Greedy Search, Genetic Algorithm and Greedy Hill Climbing are the popular search algorithms. The common objective of these algorithms is how to reduce the search space.
In this step, we first learn the biclusters produced from the previous step using Greedy Hill Climbing search algorithm and BDe Scoring Function implemented in Biolearn [23] at Department of Biological Sciences, Columbia University. For examples, for the 219 biclusters generated by ISA algorithm (See Table 1), learning these biclusters will produce 219 subnetworks. Integrating or merging these subnetworks produces the whole network from ISA algorithm.

2.6. Network Analysis and Validation

Now after inferring the interactions between genes, how to assess if these relationships are biologically exists. It is consuming time and money to experimentally validate the full set of predictions. During the last decay the interaction databases are growing exponentially. It was reported more than 230 web-accessible biological pathway and network databases www.pathguide.org. These large databases are very promising to assist the GRN inference and to validate the inferred networks.
These interaction databases use different identifiers to identify the same gene (GI, SwissProt, internal identifiers, etc.) which is requiring the resolution of synonymous names/IDs across databases. So, we want to integrate molecular interactions and other types of high-throughput data from different public databases to build biological networks automatically. For this purpose we used BioNetBuilder [24] which is an open-source client-server Cytoscape plug-in that offers a user-friendly interface to create biological networks integrated from several databases. For example, BioNetBuilder client-server [24] retrieved for *S. Cerevisiae* more than 100,000 interactions from different databases as follows:
(BIND,16244);(BioGrid,99485);(DIP,17465);(IntAct;14331);(Interologger,5395);
KEGG,5478);(MINT,11907); numbers here represent the number of interactions per each corresponding database.
Although the network retrieved by BioNetBuilder is still incomplete we consider it as gold standard network which we will compare with.
In addition, we have to compare our algorithm's performance via previous methods. In this paper we compare our algorithm with Friedman algorithm. Friedman[3] developed a new framework for discovering interactions between genes based on multiple expression measurements which are capable of discovering causal relationships, interactions between genes other than positive correlation, and finer intra-cluster structure. He applied his approach to the dataset of Spellman et al.[7], containing 76 gene expression measurements of the mRNA levels of 6177 *S. cerevisiae* ORFs. Friedman network is available from http://www.cs.huji.ac.il/ nirf/GeneExpression/top800/.
As usual, receiver operator characteristic (ROC) curve and precision recall (PR) curves are commonly used for binary decision problems. We used the DREAM2 [25] evaluation script to compute area and ROC and PR curves. We have to define some important terms as follows:
TP: Number of edges present in the gold network and in the predicted network.
FP: Number of edges not present in the gold network but included in the predicted network.
FN: Number of edges present in the gold network but not in the predicted network.
TN: Number of edges not present in the gold network and also not included in the predicted network.
Definition of TPR, FPR, Recall and Precision could be found in [26].

3. Results and Discussion

By applying our software to the Spellman data set we could summarize the produced results in the following steps:

1- We applied KNN imputation algorithm[10] to Spellman dataset in order to substitute the missing data point with the nearest values.
2- All data set's genes that do not show significance changes were removed.
3- We applied spectral subtraction denoising algorithm to the dataset.
4- Using BicAT-plus toolbox [21] we applied the six biclustering algorithms (ISA[27], CC[28] ,MSBE[29], Bivisu[30], OPSM[31], SAMBA[32]) and one traditional clustering algorithms

(k-means[33]) to the Spellman dataset. The total number of produced biclusters/clusters is 683 and its statistics is shown in Table 1.

5- We run Greedy Hill Climbing search algorithm implemented in Biolearn program[23] to these biclusters and produced 683 subnetworks.

6- These subnetworks were integrated to generate the whole gene network per each biclustering algorithm. Edges count of these networks is shown in Table 2. If we merged the edges from all the biclustering/clustering algorithms, we produced a big network contains 5440 unique edges. We refer to this network as *ALL* network. Also In the bottom of this table is the number of edges of Friedman network.

Table 1
Number of output biclusters when applied BicAT-plus toolbox() to Spellman[7] dataset

Biclustering Algorithm	Number of produced Biclusters
ISA	219
OPSM	12
CC	100
Bivisu	100
SAMBA	52
K-means	100

Table 2
Number of Network Edges Generated by learning biclusters shown in Table 1 using Biolearn program [23].

Biclustering Network	Number of Edges
ISA	2258
OPSM	220
CC	590
Bivisu	1515
SAMBA	1611
K-means	380
ALL Total Number of unrepeated Edges	5440
Friedman	947

Figure 3 shows the performance of the biclustering networks via the gold network retrieved by BioNetBuilder[24] and Friedman network[3].

Inspecting Figure 3, we reveal that neither the generated networks from different bicluster algorithms nor the *ALL network* perform well.

There is an important note to be considered when interpreting the results of this comparison. First, the interactions documented are either physical or genetic. This implies that they may not be direct interactions. The precision may be lower than the actual precision since links maybe missing in the interactome databases; and the recall may be lower than the actual recall in part because some of the links reported in the interactome databases may be indirect rather than the direct[34]. Second, some presently unsupported edges in the constructed network may find experimental evidence in the future. Therefore, these unsupported edges are not necessarily false ones[35].

For the above reasons, the False Positive (FP) edges could be considered as True Positive (TP) if it has evidence in the interaction databases (gold network). for example if the inference network include edge between gene1 and gene3 which does not exists in gold network and if these two genes connected indirectly via another intermediate gene like gene2 we can now consider the edge between gene1 and gene3 as true positive edge.

Figure 4 shows the networks performance improvement after taking into consideration the above evaluation modification. Furthermore they show how almost the false positive edges in these networks have evidence in the gold network.
Also it should be mentioned that as we expected the sparseness nature of gene regulatory network, make using biclustering techniques(ISA, SAMBA, Bivisu) outperform the performance of the Friedman network. This will open the usage of biclustering algorithms to overcome the problem of dimensionality in the GRN inference problem.

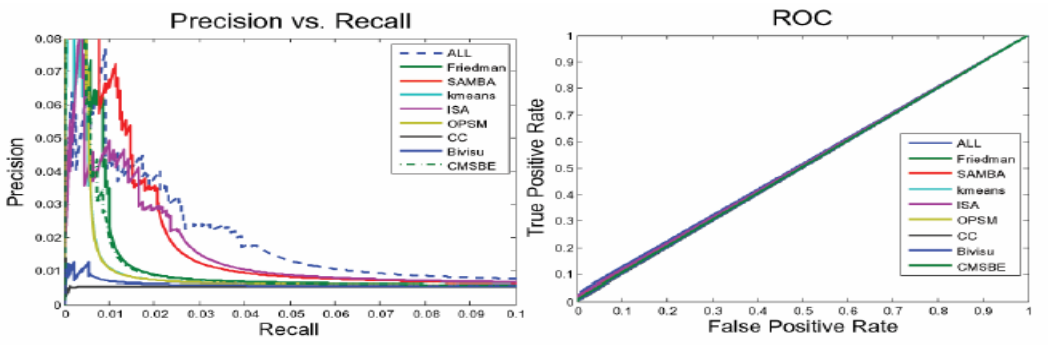

Fig. 3. ROC and PR Curves of Different Biclustering Networks

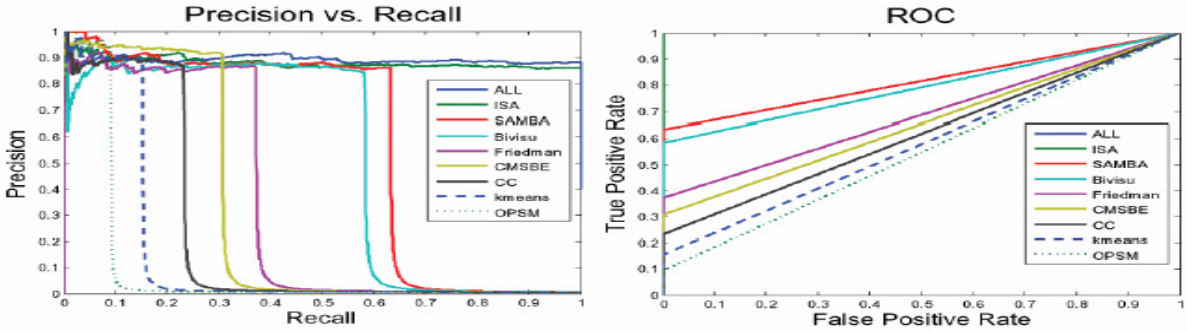

Fig. 4 ROC and PR Curves of Biclustering Networks using Modified Evaluation Methodology

4. Conclusions

The increasing development of high throughput technology like microarray, promotes researchers to study the complexity of gene regulatory network (GRN) in biological cells. GRN inference algorithms have much impact in drug development and in understanding disease ontology. Many GRN inference algorithms that are base on genome-wide data have been developed to unravel the complexity of gene regulation. Transcriptomic data measured by genome-wide DNA microarrays are traditionally used for GRN modelling. This is because RNA molecules are easily accessible in comparison to proteins and metabolites. One of the major problems with microarrays is that a dataset consists of relatively few time points with respect to a large number of genes. The dimensionality and high degree of noise are interesting problems in GRN modelling. The most common and important design rule for modelling gene networks is that their topology should be sparse. This means that each gene is regulated by only a small number of other genes. In this work a new gene regulatory network (GRN) construction system from microarray large dataset and prior biological information was proposed. As we expected the sparseness nature of GRN make biclustering techniques to show significance results compared to Friedman network[3]. In this paper we show the impact of using biclustering algorithms in GRN construction. A sophisticated filtration procedure (data filtration, missing values imputation, normalization, discretization) were used to reduce the number of expression profiles to some subset that contains the most significant genes. Also, we used our novel denoising method (Spectral

Subtraction) which accurately may account for the low SNR and able to suppresses random noise or removes some of its components. It is clear from comparison SS with previous denoising methods like Multi-Wavelet that the spectral subtraction denosing method outperforms the Multi-Wavelet method and offering a substantial improvement of the SNR. Also, The Biclustering comparison toolbox (BicAT-Plus) implemented in this paper confirms that the bicluster and cluster algorithms can be considered as integrated modules; there is no certain algorithm that can recover all the interesting patterns, what algorithm A success to recover in certain data sets, Algorithm B might fail, and vice verse. We can identify the highly enriched bi/clusters of the whole compared algorithms, integrating them to solve the dimensionality reduction problem of the gene regulatory network construction. Surprisingly, the generated networks from this study shows sufficient accuracy when comparing it via previous works and existing biological databases like BIOGRIDE.

Acknowledgments

This work is supported with a grant from University of Science \&Technology, Yemen. The Authors like to thank Prof Dana Pe'er, Columbia University; Dr Kevien Yip, Yale University and Prof G.Stolovitzky, IBM Computational Biology Center for their respectful discussion. Also we thank the lab at Stanford for making microarray data available, and the lab members for the courteous help they gave us.

References

[1] T. Ronald C., S. Mudita, W. Jennifer, K. Saeed, S. Liang, and M. Jason, "A Network Inference Workflow Applied to Virulence-Related Processes in Salmonella typhimurium," *Ann N Y Acad Sci.,* vol. 1158, pp. 143-58, 2009.

[2] S. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature,* vol. 224, pp. 177-178, 1969.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian Networks to Analyze Expression Data," *Journal of Computational Biology,* vol. 7, pp. 601-620, 2000.

[4] C. Wolfe, I. Kohane, and A. Butte, "Systematic survey reveals general applicability of ``guilt-by-association" within gene coexpression networks," *BMC Bioinformatics,* vol. 6, p. 227, 2005.

[5] T. Chen, L. H. Hongyu, and G. M. Church, "Modeling gene expression with differential equations," in *4th Pacific Symposium on Biocomputing* Big Island of Hawaii, Hawaii, USA, 1999.

[6] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear modeling of mRNA expression levels during CNS development and injury," in *4th Pacific Symposium on Biocomputing*, 1999, pp. 41-52.

[7] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. a. E. Anders, M. B. a. Brown, P. O., B. David, and B. Futcher, "Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization," *Mol. Biol. Cell,* vol. 9, pp. 3273-3297, 1998.

[8] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown, "Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes," *Mol. Biol. Cell,* vol. 11, 2000.

[9] S. K. Isaac, K. Alvin, and J. B. Atul, *Microarrays for an Integrative Genomics*: The MIT Press, 2003.

[10] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics,* vol. 17, pp. 520-525, 2001.

[11] S. V. Vaseghi, *Advanced Signal Processing and Digital Noise Reduction*: Wiley, 1996.

[12] P. D'haeseleer, "Data Requirements for Inferring Genetic Networks from Expression Data," in *4th Pacific Symposium on Biocomputing* Big Island of Hawaii, 1999.

[13] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church, "Systematic determination of genetic network architecture," *Nature Genetics,* vol. 22, pp. 281-285, 1999.

[14] R. Guthke, U. Moller, M. Hoffmann, F. Thies, and S. Topfer, "Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection," *Bioinformatics,* vol. 21, pp. 1626-1634, 2005.

[15] D. Reiss, N. Baliga, and R. Bonneau, "Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks," *BMC Bioinformatics,* vol. 7, p. 280, 2006.

[16] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics,* 2006.

[17] S. Madeira and A. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Trans Comput Biol Bioinform,* vol. 1, pp. 24-45, 2004.

[18] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo, "Validating clustering for gene expression data," *Bioinformatics,* vol. 17, pp. 309-318, 2001.

[19] F. Azuaje, "A cluster validity framework for genome expression data," *Bioinformatics,* vol. 18, pp. 319-320, 2002.

[20] S. Datta and S. Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data," *Bioinformatics,* vol. 19, pp. 459-466, 2003.

[21] F. M. Al-Akwaa, M. H. Ali, and Y. M. Kadah, "BicAT_Plus: An Automatic Comparative Tool For Bi/Clustering of Gene Expression Data Obtained Using Microarrays," in *26th National Radio Science Conference (NRSC)* Cairo, Egypt, 2009.

[22] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler, "BicAT: a biclustering analysis toolbox}," *Bioinformatics,* vol. 22, pp. 1282-1283, 2006.

[23] P. e. Dana, "Bayesian Network Analysis of Signaling Networks: A Primer," *Sci. STKE,* vol. 2005, pp. pl4-, 2005.

[24] I. Avila-Campillo, K. Drew, J. Lin, D. J. Reiss, and R. Bonneau, "BioNetBuilder: automatic integration of biological networks," *Bioinformatics,* vol. 23, pp. 392-393, 2007.

[25] G. Stolovitzky, R. Prill, and A. Califano, "Lessons from the DREAM2 Challenges," *Annals of the New York Academy of Sciences,* pp. 1158:159-95, 2009.

[26] D. Jesse and W. Madison, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning*: ACM New York, NY, USA, 2006, pp. 233 - 240.

[27] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai, "Revealing modular organization in the yeast transcriptional network," *Nature Genetics,* vol. 31, pp. 370-377, 2002.

[28] Y. Cheng and G. Church, "Biclustering of expression data," in *Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology*, 2000, pp. 93-103.

[29] X. Liu, L. Wang, and A. Takaluoma, "Computing the maximum similarity bi-clusters of gene expression data," *Bioinformatics,* vol. 23, pp. 50-56, 2007.

[30] K. Cheng, N. Law, W. Siu, and T. Lau, "BiVisu: software tool for bicluster detection and visualization," *Bioinformatics,* vol. 23, pp. 2342-2344, 2007.

[31] A. a. C. Ben-Dor, B and Karp, R and Yakhini, Z},, "Discovering local structure in gene expression data: the order-preserving submatrix problem," *Journal of Computational Biology,* vol. 10, pp. 373-384, 2003.

[32] A. Tanay, R. Sharan, and R. Shamir, "Discovering statistically significant biclusters in gene expression data," *Bioinformatics,* vol. 18, pp. S136-144, 2002.

[33] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," in *Proceedings of the National Academy of Sciences of the United States of America*, 1998, pp. 14863-14868.

[34] A. C. Lozano, N. Abe, Y. Liu, and S. Rosset, "Grouped graphical Granger modeling for gene expression regulatory networks discovery," *Bioinformatics,* vol. 25, pp. i110-118, 2009.

[35] X.-w. Chen, G. Anantha, and X. Wang, "An effective structure learning method for constructing gene networks," *Bioinformatics,* vol. 22, pp. 1367-1374, 2006.