Supervised Inference of Gene Regulatory Networks by Linear Programming

Yong Wang[1,2], Trupti Joshi[3], Dong Xu[3], and Xiang-Sun Zhang[2], and Luonan Chen[1,4]

[1] Osaka Sangyo University, Nakagaito 3-1-1, Daito, Osaka 574-8530, Japan
ywang@ctex.org, chen@elec.osaka-sandai.ac.jp
[2] Academy of Mathematics and Systems Science, CAS, Beijing 100080, China
zxs@amt.ac.cn
[3] Computer Science Department and Christopher S. Bond Life Sciences Center, University of Missouri, Columbia, MO 65211, USA
{joshitr, xudong}@missouri.edu
[4] Institute of systems biology, Shanghai University, 200444, China

Abstract. The development of algorithms for reverse-engineering gene regulatory networks is boosted by microarray technologies, which enable the simultaneous measurement of all RNA transcripts in a cell. Meanwhile the curated repository of regulatory associations between transcription factors (TF) and target genes is available based on bibliographic references. In this paper we propose a novel method to combine time-course microarray dataset and documented or potential known transcription regulators for inferring gene regulatory networks. The gene network reconstruction algorithm is based on linear programming and performed in the supervised learning framework. We have tested the new method using both simulated data and experimental data. The result demonstrates the effectiveness of our method which significantly alleviates the problem of data scarcity and remarkably improves the reliability.

Keywords: Systems biology, gene regulatory network, linear programming.

1 Introduction

Microarray technologies have produced tremendous amounts of gene expression data [1]. For example the Stanford Microarray Database (SMD) has deposited data for 60,222 experiments, from 302 labs and 36 organisms, as of March, 2006. It is necessary and important to understand gene expression and regulation through mining these data. A straightforward way on microarray data analysis is the reconstruction of gene regulatory network, which aims to find the underlying network of gene-gene interactions from the measured dataset of gene expression [2]. A wide variety of approaches have been proposed to infer gene regulatory networks from time-course data [3, 4, 5] or perturbation experiments [6], such as discrete models of Boolean networks and Bayesian networks, and

D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNBI 4115, pp. 551–561, 2006.

continuous models of neural networks, difference equations [1] and differential equations [7, 8].

The common problem for all these models is scarcity of data [9]. Since a typical gene expression dataset consists of relatively few time points (often less than 20) with respect to a large number of genes (generally in thousands). In other words, the number of genes far exceeds the number of time points for which data are available, making the problem of determining gene regulatory network structure a difficult and ill-posed one.

In this paper we propose a novel method to combine computational analysis of microarray dataset and biological experiment results together for inferring gene regulatory network with the consideration of sparsity of connections. We develop a supervised gene network reconstruction algorithm by linear programming based on the differential equation model [9]. The original idea of supervised learning is to incorporate the known documented interactions between genes into the parameter estimation by keeping sparsity, because many regulatory associations have been identified and recorded in literatures or databases, which are valuable information for the inference of gene networks. For example, YEASTRACT (Yeast Search for Transcriptional Regulators And Consensus Tracking, `http://www.yeastract.com/`) is a curated repository of more than 12,500 regulatory associations between transcription factors (TF) and target genes in *Saccharomyces cerevisiae*, based on more than 900 bibliographic references. All the information in YEASTRACT will be updated regularly to match the recent literature on yeast regulatory networks. In this paper, the supervised information is adopted in the inference procedure by exploiting the general solution form of arbitrary Jacobian matrix for gene regulatory network. The proposed method theoretically ensures the derivation of feasible network structure with respect to the used dataset, thereby not only significantly alleviating the problem of data scarcity but also remarkably improving the reliability. Specifically, it can be expected that the information of documented regulatory interactions considered will lead to biologically plausible results.

2 Methods

In general, one can represent a gene regulatory network model as a directed graph. In this paper the graph is mathematically expressed by a set of linear differential equations. The regulatory influence between genes are formulated as a matrix and obtained as the general solution of differential equations [9]. The supervised learning information is added in the procedure of seeking general solution from particular solution by solving a linear programming problem. Fig. 1 illustrates the schematic of the proposed method. The experiment data obtained by microarray technology is analyzed and normalized, then the time course data of gene expression are collected as a matrix. The dynamic properties of the gene regulatory network are described by ordinary differential equation model. To infer the relationships between genes, previously known regulatory interactions in the network are picked as supervised information. Then novel influences (or

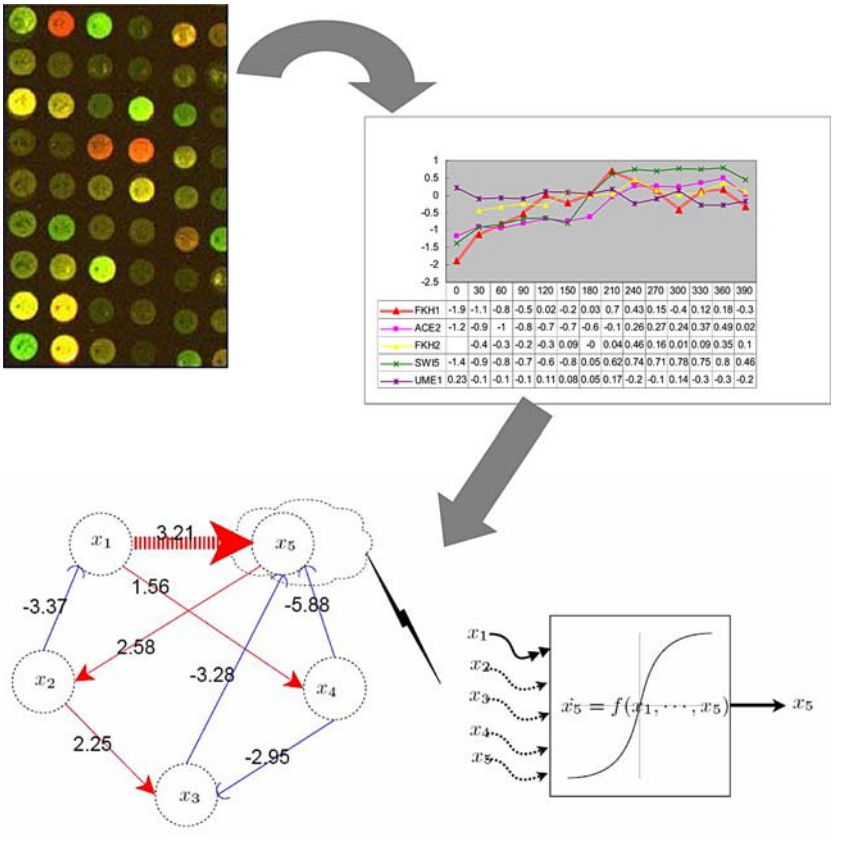


Fig. 1. Graph depiction of the strategy to construct the network by supervised learning. The experimenter measures the expression (concentration) of many or all RNA transcripts in the cells by microarray. Then time course data are collected as a matrix by normalization. The ordinary differential equation is used to inference the relationship between genes. In the network previously known regulatory influences are marked in bold and special line, novel influences (or false positives) are marked in common line. Arrows and arcs denote activation and repression, respectively.

false positives) are obtained by linear programming inference algorithm as a result. Noticed that the network structure inferred with or without supervision are all solutions of the ordinary differential equation. They differ only in that the supervised one learns the known correct information in the inference procedure and the results tend to be more consistent with the known literatures.

2.1 Gene Regulatory Network

The model is based on relating the changes in gene transcript concentration to each other and to the external perturbation [10]. The external perturbation means an experimental treatment that can alter the transcription rate of the genes in the cell. An example of perturbation is the treatment with

a chemical compound, or a genetic perturbation involving over-expression or down-regulation of particular genes. The ordinary differential equation is used to represent the rate of synthesis of a transcript as a function of the concentrations of every other transcript in a cell and the external perturbation:

$$\dot{x}(t) = Jx(t) + pc(t), \quad t = t_1, ..., t_m \tag{1}$$

Where $x(t)$ is expression level (mRNA concentrations) of gene at time point t, $J = (J_{ij})_{n\times n} = \partial f(x)/\partial x$ is an $n \times n$ Jacobian matrix or connectivity matrix. Here p represents the effort of perturbation on x and $c(t)$ represents the external perturbation at time t. By introducing p as a variable in inference, the approach can be a powerful methodology for the drug discovery process. Since it would be able to identify the compound mode of action via a time course gene expression profile and the feasibility is proved by reconstruct the SOS system in the bacteria *Escherichia coli* [6, 10].

Writing the equation (1) in a compact form for all time points using matrix notation as

$$\dot{X} = JX + PC \tag{2}$$

where $X = (x(t_1), ..., x(t_m))$ and $\dot{X} = (\dot{x}(t_1), ..., \dot{x}(t_m))$ are all $n \times m$ matrices with the first derivative of mRNA concentration $\dot{x}_i(t_j) = [x_i(t_{j+1}) - x_i(t_j)]/[t_{j+1} - t_j]$ for $i = 1, ..., n; j = 1, ..., m$. Suppose that there are s times external perturbation, then $C = (c(t_1), ..., c(t_m))$ is a $s \times m$ matrix representing the s perturbations. The unknowns to calculate are connectivity matrix J and P. J is an $n \times n$ connectivity matrix, composed of elements J_{ij}, which represents the influence of gene j on gene i with a positive, zero or negative sign indicating activation, no interaction and repression respectively. P is an $n \times s$ matrix representing the effect of s perturbations on the n gene system. An non-zero element P_{ij} of P implies that the ith gene is a direct target of the jth perturbation. The equation (2) can be reformed as:

$$\dot{X} = [J \ P] \begin{bmatrix} X \\ C \end{bmatrix} \tag{3}$$

By adopting SVD to $\begin{bmatrix} X \\ C \end{bmatrix}$, i.e.,

$$\begin{bmatrix} X \\ C \end{bmatrix}^T_{m\times(n+s)} = U_{m\times(n+s)} E_{(n+s)\times(n+s)} V^T_{(n+s)\times(n+s)} \tag{4}$$

where U is a unitary $m \times (n + s)$ matrix of left eigenvectors, $E = diag(\ e_1, \cdots, e_{(n+s)})$ is a diagonal $(n + s) \times (n + s)$ matrix containing the $(n + s)$ eigenvalues and $(V)^T$ is the transpose of a unitary $(n + s) \times (n + s)$ matrix of right eigenvectors. Then we can have a particular solution with the smallest L_2 norm for the Jacobian matrix $\hat{J} = (\hat{J}_{ij})_{n\times n}$ and $\hat{P} = (\hat{P}_{ij})_{n\times s}$ as

$$[\hat{J} \ \hat{P}] = (\dot{X})U(E)^{-1}V^T \tag{5}$$

where $E^{-1} = diag(1/e_i)$ and $1/e_i$ is set to be zero if $e_i = 0$. Thus, the general solution of the Jacobian matrix $J = (J_{ij})_{n\times n}$ and $P = (P_{ij})_{n\times s}$ are

$$[J\ P] = [\hat{J}\ \hat{P}] + YV^T \tag{6}$$

$Y = (y_{ij})$ is an $n \times (n + s)$ matrix. Solutions of (6) represent all of the possible networks that are consistent with the perturbation microarray dataset, depending on arbitrary Y. Then given the gene of interest, the supervised information can be incorporated during the process of getting the sparse structure of J by similarly solving linear programming as described in 2.2.

2.2 Supervised Learning by Linear Programming

With the general solution expression of (6), the next step is to pick a biologically meaningful solution by determining variable Y. In [11], the objective is to make the zero elements of J as much as possible. Though the inferred network is sparse in some sense, it is still a heuristic idea and cannot be biologically ensured. In this paper we add the supervised information during the inference process of the network structure. This strategy will drive the network toward the direction in a more biological meaning way.

Suppose that the known information of gene regulatory network is expressed by K, which is an $n \times n$ sparse matrix. If the element K_{ij} is nonzero, it means that gene j has regulatory influence (the activation or depression depends on the sign of K_{ij}) on gene i and this interaction is assumed to be revealed by biological experiments. We will discuss how to incorporate these valuable information in the inference of whole network by linear programming formulation.

The $\hat{J}$ is the particular solution of the microarray dataset by SVD. The information of K is incorporated in the general solution $J = (J_{ij})_{n\times n}$ by a proper Y such that

$$[J\ P] = [\hat{J}\ \hat{P}] + YV^T, \tag{7}$$

and at the same time the following conditions are satisfied

$$(J_{ij})_{n\times n} = (K_{ij})_{n\times n}, \quad K_{ij} \neq 0 \tag{8}$$

Considering the L_1 problem:

$$\min_{Y} |[(K - \hat{J})\ \ \hat{P}] + Y^k V^T| \tag{9}$$

where $\hat{J}$, K, $\hat{P}$ and V^T are given. Without loss of generality, the problem is expressed as the standard form

$$\min_{X} |AX - B| \tag{10}$$

where the coefficient A is an $n \times (n + s) \times nl$ matrix, and l is the number of zero elements in E^{-1} in (4). X and B are $l \times n$ and $(n + s) \times n$ matrices, respectively. In fact we Do not need to solve such a large LP problem. Observing that A has a special structure as

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \cdots & \\ & & & A_n \end{bmatrix}$$

The size of $A_i, i = 1, 2, \cdots, n$ is $(n + s) \times l$. With the correspondingly decomposing $X = [X_1, X_2, \cdots, X_n]$ and $B = [B_1, B_2, \cdots, B_n]$, the solution of the raw problems (10) can be obtained by solving the following n small problems:

$$\min_{X_i} |A_i X_i - B_i| \quad i = 1, 2, \ldots, n \tag{11}$$

where the dimensions of A_i, X_i and B_i are $n \times l$, $l \times 1$ and $n \times 1$, respectively. Clearly the decomposition scheme reduces the storage space from about $O(n^4)$ to $O(n^2)$ and requires $O(n^4)$ computations. Furthermore, it provides the formulation for parallel computation so as to rapidly find regulatory relationship of the interested gene in a high priority. i. e. the regulatory influence relationship of a designated gene can be obtained independently by solving a small scale L1 problem. Without loss of generality, only gene i is to be considered in the following inference algorithm.

Noticing that $B_i = -\hat{J}_i + K_i (B_i,\ \hat{J}_i$ and K_i are the ith row of the matrix B, $\hat{J}$ and K, respectively), let $I = \{j | K_{ij} \neq 0\}$ and $|I| = q$, by introducing $2(n - q)$ slack variables $u_j, v_j,\ j \in \{1, 2, \cdots, n\} \setminus I$, the L1 problem

$$\min_{x_{ij}} \textstyle\sum_{j=1}^{n} | \sum_{k=1}^{l} a_{ik} x_{ik} - \hat{J}_{ij} + K_{ij}| \tag{12}$$

is equivalent to the linear programming model as follows:

$$\begin{aligned} \min_{x_{ij}, u_j, v_j} \quad & \textstyle\sum_{j \in I} (u_j + v_j) \\ s.t. \quad & \textstyle\sum_{k=1}^{l} a_{ik} x_{ik} - \hat{J}_{ij} + K_{ij} = u_j - v_j, \ j \in \{1, 2, \cdots, n\} \setminus I \\ & \textstyle\sum_{k=1}^{l} a_{ik} x_{ik} = \hat{J}_{ij}, \qquad j \in I \\ & u_j, v_j \geq 0, \qquad j \in \{1, 2, \cdots, n\} \setminus I \\ & x_{ij} \in R, \qquad j \in \{1, 2, \cdots, n\} \end{aligned} \tag{13}$$

The variables need to be solved are $x_{i1}, x_{i2}, \cdots, x_{il}, u_1, v_1, \cdots, u_{n-q}, v_{n-q}$ and the total number is $2n - 2q + l$. There are n equality constraints and $2(n - q)$ inequality constraints. The LP problem for such a scale can be dealt directly by simplex method.

In the above linear programming formulation, the supervised information is incorporated through the first $n-q$ equality constraints. If a little supervised information is added (q is small), the LP will generally give the unique solution which is a particular solution and satisfies $(J_{ij})_{n \times n} = (K_{ij})_{n \times n}, \quad K_{ij} \neq 0$. But when much supervised information is applied to the gene regulatory system (q is large), the LP will give an approximate solution due to redundant constraints. In this case the system is over-determined and not all $(J_{ij})_{n \times n} = (K_{ij})_{n \times n}, \quad K_{ij} \neq 0$ are satisfied. The answer of how much supervised information should be incorporated in the inference depends on the inherent degree of freedom depicted by

l. For example in a large-scale gene network, the time course data is relatively scarce ($l \approx n$) and much supervised information can be added. As reported in [11] for a large system, the smallest number of time points needed is $O(\log n)$ to reconstruct the $n \times n$ connectivity matrix for an n-gene network. Hence, the proper incorporation of known information of interactions in the inference algorithm will alleviate the requirement and dependence on high quality microarray data greatly.

Next we want to further address how to set values for matrix K from know information. In our model the knowledge of gene regulatory network is expressed and incorporated by matrix K. If the element K_{ij} is nonzero, it means that gene j has regulatory influence on gene i. The activation or depression role differs on the sign of K_{ij}. It is better to provide the quantitative strength of the know regulatory interactions, but many constraints are basically all qualitative instead of quantitative in the databases or literatures though it is true that they are readily available . You may know gene i activates gene j, but the quantitative relationship as described by the K_{ij} is not known. The feasible way is to assign a constant to K_{ij} and use its sign to indicate activation or depression relationship. Since in some meaning, the major function of element k_{ij} is to introduce constraints in the linear programming and this function can be preformed by keeping it a constant. In the long run, a systematic search/protocol or a soft (boundary) constraint can be developed. In this paper, The $k_{ij} = 1$ or $k_{ij} = -1$ are simply set to know activation or regression regulatory interaction in our experiments using biological microarray data.

3 Results

In this section, we first report on simulated numerical test that we have designed to benchmark our method by using supervised inference strategy. Then we will describe the gene regulatory network inference using yeast microarray gene expression data. As analyzed in Methods section, when no supervised information is applied, our method is similar to the method of [11], which can recover the network connectivity from gene expression measurements in the presence of noise by singular value decomposition (SVD) and regression. With supervised information, we can further infer the network structure in a more accurate and robust manner. In this section, only preliminary results are given for incorporating supervised information by single microarray dataset, the experiments of gene network inference by perturbation dataset and multiple datasets can be conducted similarly.

3.1 Simulated Data

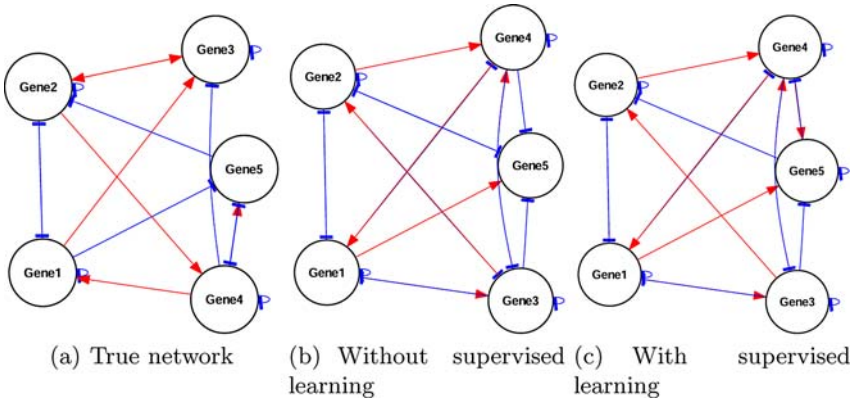
The first example is a small simulated network with five genes governed by

$$\begin{aligned}
\dot{x}_1(t) &= -x_1(t) - 0.2x_2(t) + 0.5x_4(t) + \xi_1(t),\\
\dot{x}_2(t) &= -0.8x_1(t) - 1.5x_2(t) + x_3(t) - 0.5x_5(t) + \xi_2(t),\\
\dot{x}_3(t) &= 0.6x_1(t) + 0.2x_2(t) - x_3(t) - 0.3x_4(t) + \xi_3(t),\\
\dot{x}_4(t) &= 0.9x_2(t) - x_4(t) - 1.5x_5(t) + \xi_4(t),\\
\dot{x}_5(t) &= -0.2x_1(t) + 0.7x_4(t) - 1.5x_5(t) + \xi_5(t),
\end{aligned}$$

where x_i reflects the expression level of the gene-i and $\xi_i(t)$ represents noise for $i = 1, 2, 3, 4, 5$.

To test our method, we randomly choose the initial condition of the system and take several points of x as a measured time-course dataset. In our simulated example, we obtained a dataset with 4 time points, and applied our method to reconstruct the connectivity matrix or the Jacobian matrix J. We set all of noises $\xi_i(t), i = 1, 2, 3, 4, 5$ obeying normal distribution in the simulated example with noise level of $N(0, 0.005)$. And the supervised information K is incorporated by fixing $K_{2,5} = -0.5$ and $K_{5,4} = 0.7$, which means the depression influence of gene 5 $\rightarrow$ gene 2 and the activation influence of gene 4 $\rightarrow$ gene 5 are measured and known.

The numerical results are depicted in Figure 2, which shows the reconstructed networks without and with supervised information, respectively. Clearly with the supervised information, it infers the network more accurately. Without supervised information (Fig. 2 (b)), the strong self repressive relation of gene x_5, the activation relation between gene x_4 and gene x_5 are neglected. When the supervised information is used, the topology of the network becomes correct and the predicted connectivity matrix, which represents the strengths among gene interactions, is very close to the true one (Fig. 2 (c)). Such results imply that our method is able to infer the solution of the highly under-determined problem in an accurate manner when a sufficient number of known interactions are available even although the microarray dataset has only a few time points.



(a) True network (b) Without supervised learning (c) With supervised learning

Fig. 2. Regulatory network reconstruction for the simulated example with 5 genes. Activation is shown in red arrow and repression in blue arc.

3.2 Application to Experimental Data

We applied our method using experimental data. To ensure high quality of the data, we only used microarray experimental data generated from whole genome Affymetrix chips, instead of any oligo or cDNA array data. The experiments are conducted to test our method using a small number of genes of Heat-Shock

Response data for yeast. We created an input dataset for 10 transcription factors related to heat-shock response in yeast *Saccharomyces cerevisiae*. 2 out of the 10 transcription factors (HSF1 and SKN7) are known to be directly involved in heat shock response. HSF1 and SKN7 each are known to regulate 4 other transcription factors among the ten. TYE7 → HSF1 and RPN4 → HSF1 are documented known regulation. This information was obtained from YEASTRACT (`http://www.yeastract.com/index.php`). For the 10 transcription factors, we used microarray dataset at the Stanford Microarray Database (`http://smd.stanford.edu/`) (y14, with 5 time points) for gene expression under heat shock conditions. We applied the proposed method to this dataset. As indicated in 2.2 subsection, the $k_{ij} = 1$ or $k_{ij} = -1$ are simply set to know regulatory interaction in our real data experiments. As shown in Fig. 3, the prediction succeeded in reconstructing 2 edges of the network which are identical with the documented known regulation (see subfigure (b) in Fig. 3). But these regulation information can not be obtained without the supervised learning procedure (see the subfigure (a) in Fig. 3).

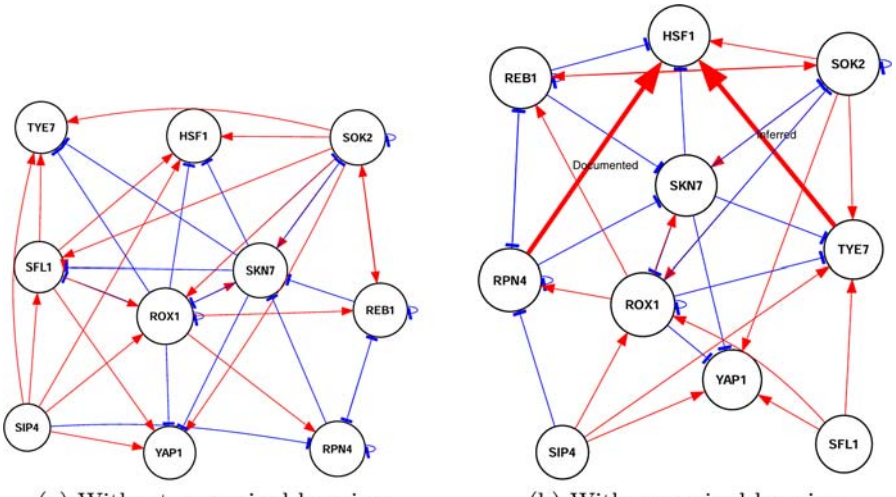


Fig. 3. Regulatory network reconstruction for a set of 10 transcription factors for heat shock response microarray data y14. Activation is shown in red arrow and repression in blue arc. The confirmed edges are shown in bold arrows with labels.

4 Discussion and Conclusion

Microarray gene expression data become increasingly common data source that can provide insights into biological processes at a system-wide level. In contrast to the conventional methods suffering the dimensionality problem, the main contribution of this paper is development of a methodology to reconstruct gene regulatory network by using existing interaction information of genes from database.

In other words, we provide a general framework to handle the microarray data by fully considering the accumulative biological experiment results, which not only makes the inferred results more accurate, but also alleviates the problem of dimensionality or data scarcity greatly. We have tested our approach to both simulated small-size problems and experimental biological data in this paper, which verified the efficiency and effectiveness of our algorithm.

In this paper, the supervised learning strategy is incorporated in the network inference. These approaches are called "supervised" because the model is learned from a training data consisting of a set of system responses. In our method, the training data are collected from the known transcription relationship from YEASTRACT, which allows the identification of documented or potential transcription regulators of a given gene and documented or potential regulatory for each transcription factor. Compared with the supervised learning strategy in [6], our method has advantages in simplifying the design of experiments and requiring less expensive computation. In our method, only time-course data of gene expression is utilized. While in [6] the perturbations around steady state are used. The reverse engineering algorithm implemented by linear programming in our method simplifies the complexity of network inference greatly. The strategy in [6] to fix the number of interaction of every gene and to enumerate all the cases by greedy algorithm is expensive and time consuming compared with the simplex algorithm for linear programming.

To examine causal relation among genes, a major source of errors comes from the noises of the gene expression data intrinsic to microarray technologies. To reduce the defect of unreliable data, a feasible method is to combine multiple microarray gene expression datasets together to get the more consistent network. Our supervised strategy in inferring gene regulatory network is ready to generalized in multiple datasets case. The more simulations and experiments are in progress.

The tendency is clearly that more and more data of gene interaction will be experimentally measured, reported and deposited in the literatures or databases. Meanwhile the quality of these data and microarray data keeps improving. It is reasonable to expect that our inference method will continue to prove valuable in analyzing and predicting the behavior or gene regulatory networks.

References

1. van Someren, E. P., Wessels, L. F. A., Reinders, M. J. T., Backer, E.: Robust Genetic Network Modeling by Adding Noisy Data. In: Proceedings of 2001 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, Japan, (2001) 234-246
2. Hartemink, A. J.: Reverse Engineering Gene Regulatory Networks. Nature Biotechnology, 23 (5) (2005) 554-555
3. Holter, N. S., Maritan, A., Fedoroff, M. C. N. V., Banavar, J. R.: Dynamic Modeling of Gene Expression Data. Proc. Natl. Acad. Sci. USA, 98 (2001) 1693-1698
4. Tegner, J., Yeung, M. K. S., Collins, J.: Reverse Engineering Gene Networks: Integrating Genetic Perturbations with Dynamical Modeling. Proc. Natl. Acad. Sci. USA, 100 (2003) 5944-5949

5. Dewey, T. G., Galas, D. J.: Dynamic Models of Gene Expression and Classification. Functional & Integrative Genomics, 1 (4) (2001) 269-278
6. Gardner, T. S., Di Bernardo, D., Lorenz, D., Collins, J. J.: Inferring Genetic Networks and Identifying Compound Mode of Action Via Expression Profiling. Science, 301 (5629) (2003) 102-105
7. Chen, L., Aihara, K.: Stability and Bifurcation Analysis of Differential-difference-algebraic Equations. IEEE Trans. on Circuits and Systems - I, 48 (3) (2001) 308–326
8. Chen, L., Aihara, K.: Stability of Genetic Regulatory Networks with Time Delay. IEEE Trans. on Circuits and Systems-I,49 (5) (2002) 602-608
9. Wang, Y., Joshi, T., Zhang, X. S., Xu, D., Chen, L.: Inferring Gene Regulatory Networks from Multiple Microarray Datasets (2006) (In submission)
10. Bansal, M., Gatta, G. D., di Bernardo, D.: Inference of Gene Regulatory Networks and Compound Mode of Action from Time Course Gene Expression Profiles. Bioinformatics, 10 (1093) (2006) 1-8
11. Yeung, M. K. S., Tegner, J., Collins, J.: Reverse Engineering Gene Networks Using Singular Value Decomposition and Robust Regression. Proc. Natl. Acad. Sci., USA, 99 (2002) 6163-6168