Reverse Engineering Yeast Gene Regulatory Networks Using Graphical Models

Jiayin Wang, Yufei Huang
Department of ECE
The University of Texas at San Antonio
San Antonio, TX 78249-0669
E-mail: yhuang@utsa.edu

Maribel Sanchez, Yufeng Wang
Department of Biology
The University of Texas at San Antonio
San Antonio, TX 78249-0669
Email: yufeng.wang@utsa.edu

Jianqiu (Michelle) Zhang
Department of ECE
University of New Hampshire
Durham, NH 03824.
Email: jianqiu.zhang@unh.edu

***Abstract*— We investigate in this paper reverse engineering of gene regulatory networks from time series microarray data. We propose a dynamic Bayesian networks (DBNs) modeling and a full Bayesian learning scheme. The proposed DBN models directly the continuous expression levels and also is associated with parameters that indicate the degree as well as the types of regulations. To learn the network from data, we proposed a reversible jump Markov chain Monte Carlo (RJMCMC) algorithm. The RJMCMC algorithm can provide not only more accurate inference results than the deterministic alternative algorithms but also an estimate on the *a posteriori* probabilities (APPs) of the network topology. The estimated APPs provide useful information on the confidence of the inferred results and can also be used for efficient Bayesian data integration. The proposed approach was tested on yeast cell cycle microarray data and the results were compared with the KEGG pathway map.**

I. Introduction

We study in this paper signal processing solutions to the inference of genetic regulatory networks (GRNs) based on microarray data. A gene regulatory network is a network representing regulations between genes in a cell. Genes are nodes in this network and edges are regulatory relationships between genes. GRNs are the regulatory circuits linking proteins and targets. They are of great importance to elucidate the system structure, system dynamics, control molecular systems. Microarray data provide first-hand information on genome wide molecular interactions and inference of GRNs based on microarray data is referred to as "reverse engineering" [1]. The solution to this problem is complicated by the enormously large scale of the unknowns and rather small sample size, not to mention the inherent experimental defects and many other factors. Of interest to this paper are the time series microarray data, which reflect the dynamics of gene regulation in cell cycles.

In this paper, we apply dynamic Bayesian networks (DBNs) to model the time series microarray experiment. The DBN used in this paper is close to that in [2] [3], which models the continuous expression level and the degree of regulation. However, unlike in [2], we target cases where only microarray data are available for network inference. Consequently, a more conservative linear regulatory model is adopted as in [3] [4] since more complex models will greatly reduce the credibility of the inferred results. Also we are interested in full Bayesian solutions for learning the networks. That can provide estimates on the *a posteriori* probabilities (APPs) of the inferred network topology. In the context of GRNs, the APPs provide valuable measurements of confidence on inference. To this end, we propose a solution based on reversible jump Markov chain Monte Carlo (RJMCMC) sampling.

The rest of the paper is organized as follows: In section II, the modeling issues on the time series data with DBNs are discussed. In section III, tasks on learning the networks are formulated and the Bayesian solution is derived. In section IV, the test results of the proposed approach are provided and the conclusion is drawn.

II. Modeling with Dynamic Bayesian Networks

To model the gene regulation in cell cycles using DBNs, we assume to have a microarray that measures the expression levels of G genes at $N+1$ evenly sampled consecutive time instances. We define a random variable matrix $\mathbf{Y} \in \mathcal{R}^{G\times(N+1)}$ with the (i,n)th element $y_i(n-1)$ denoting the expression level of gene i measured at time $n-1$ (See Figure 1). We further assume that the gene regulation follows a first-order time-homogeneous Markov process. As a result, we only consider regulatory relationships between two consecutive time instance and this relationship remains unchanged over the course of the microarray experiment. We call the regulating genes as the parent genes or parents for short.

The structure of the proposed DBNs for modeling the cell cycle regulations is illustrated in Figure 1. In this DBN, each node denotes a random variable in $\mathbf{Y}$ and all the nodes are arranged the same way as the corresponding variables in the matrix $\mathbf{Y}$. An edge between two nodes denotes the regulatory relationship between the two associated genes and the arrow indicates the direction of regulation. Like all Bayesian networks, DBNs do not allow circles in the graph.

To complete modeling with DBNs, we need to define the conditional distributions of each child nodes over the graph. To define the conditional distributions, we let $\mathbf{pa}_i(n)$ denote a column vector of the expression levels of all the parent genes that regulate gene i measured at time n. As an example in Figure 1, $\mathbf{pa}_i(n)^\top = [y_1(n), y_3(n), y_G(n)]$. Then, the conditional distributions of each child nodes over the DBNs can be expressed as $p(y_i(n)|\mathbf{pa}_i(n-1))\ \forall i$. To determine

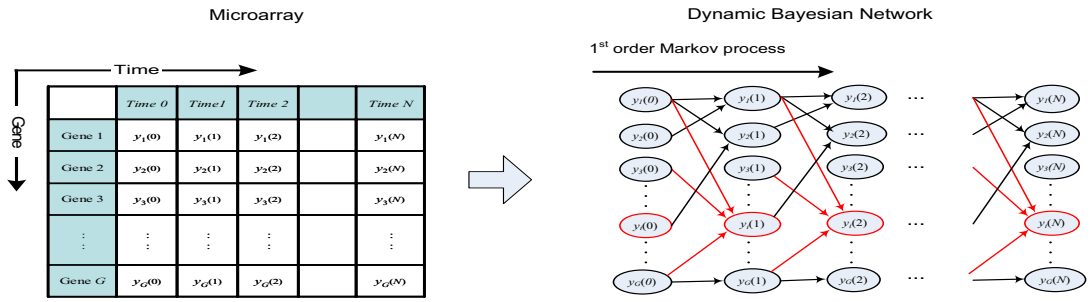

Fig. 1. A DBN modeling of time series expression data.

the expression of the distributions, we assume the expression level of gene i is the result of linear combination of the expression levels of the regulating genes at previous sample time. Mathematically, we have the following expression

$$y_i(n) = \mathbf{w}_i^\top \mathbf{pa}_i(n-1) + e_i(n), \quad n = 1, 2, \cdots, N \tag{1}$$

where $\mathbf{w}_i \in \mathcal{R}$ is the weight vector independent of time n and $e_i(n)$ is assumed to be white Gaussian noise with variance σ^2. The weight vector indicates the degree and the types of the regulation [3]. A gene is up-regulated if the weight is positive and is down-regulated otherwise. The magnitude (absolute value) of the weight indicates the degree of regulation. The noise variable is introduced to account for modeling and experimental errors. From (1), we obtain that the conditional distribution is a Gaussian distribution, i.e.,

$$p(y_i(n)|\mathbf{pa}_i(n-1)) = \mathcal{N}(\mathbf{w}_i^\top \mathbf{pa}_i(n-1), \sigma_i^2). \tag{2}$$

In (1), the weight vector $\mathbf{w}_i$ and the noise variance σ_i^2 are the unknown parameters to be determined.

III. Learning the DBN

The task of learning the above DBN consists of two parts: structure learning and parameter learning. The objective of structure learning is to determine the topology of the network or the parents of each gene. Under a given structure, parameter learning involves the estimation of the weight vector $\mathbf{w}_i$ and the noise variance σ_i^2 for all i. Since the gene expression levels at any given time are independent and the network is fully observed, we can learn the parents and the associated model parameters of each gene separately.

A. Bayesian criterion for structural learning

Let $\mathcal{M}_i = \{M_i^{(1)}, M_i^{(2)}, \cdots, M_i^{(K)}\}$ denote a set of all possible network topologies for gene i, where each element represents a topology derived from a possible combination of the parents of gene i. The problem of structure learning is to select the topology from $\mathcal{M}_i$ that is best supported by the microarray data.

We can express (1) for a particular topology $M_i^{(k)}$ in a matrix-vector form

$$\mathbf{y}_i = \mathbf{Pa}_i^{(k)} \mathbf{w}_i^{(k)} + \mathbf{e}_i^{(k)} \tag{3}$$

where

$$\begin{aligned}
\mathbf{y}_i &= [y_i(1), \cdots, y_i(N)]^\top, && (4)\\
\mathbf{Pa}_i^{(k)} &= [\mathbf{pa}_i^{(k)}(0), \mathbf{pa}_i^{(k)}(1), \cdots, \mathbf{pa}_i^{(k)}(N-1)]^\top, && (5)\\
\mathbf{e}_i^{(k)} &= [e_i^{(k)}(1), e_i^{(k)}(2), \cdots, e_i^{(k)}(N)]^\top, && (6)\\
\mathbf{w}_i^{(k)} &= [w_i^{(k)}(0), w_i^{(k)}(1), \cdots, w_i^{(k)}(N-1),]^\top. && (7)
\end{aligned}$$

We select the most probable topology $\bar{M}_i$ based on the maximum *a posterior* criterion (MAP)[5], i.e.,

$$\begin{aligned}
\bar{M}_i &= \arg \max_{M_i^{(k)} \in \mathcal{M}_i} p(M_i^{(k)}|\mathbf{Y}) \\
&= \arg \max_{M_i^{(k)} \in \mathcal{M}_i} p(\mathbf{y}_i|\mathbf{Pa}_i^{(k)}) p(M_i^{(k)}) \qquad (8)
\end{aligned}$$

where $p(\mathbf{y}_i|\mathbf{Pa}_i^{(k)})$ is the marginal likelihood and $p(M_i^{(k)})$ is the model prior.

The marginal likelihood, $p(\mathbf{y}_i|\mathbf{Pa}_i^{(k)})$, is obtained by integrating the unknown parameters from the full likelihood

$$p(\mathbf{y}_i|\mathbf{Pa}_i^{(k)}) = \int\int p(\mathbf{y}_i|\mathbf{w}_i^{(k)}, \sigma_{ik}^2, \mathbf{Pa}_i^{(k)}) p(\mathbf{w}_i^{(k)}, \sigma_{ik}^2|\mathbf{Pa}_i^{(k)}) d\mathbf{w}_i^{(k)} d\sigma_{ik}^2 \tag{9}$$

where $p(\mathbf{w}_i^{(k)}, \sigma_{ik}^2|\mathbf{Pa}_i^{(k)})$ is the parameter prior and we choose the standard conjugate Gaussian-Inverse-Gamma prior [6]

$$p(\mathbf{w}_i^{(k)}, \sigma_{ik}^2|\mathbf{Pa}_i^{(k)}) = \mathcal{N}_{\mathbf{w}_i^{(k)}}(\mathbf{0}, \sigma_{ik}^2 \mathbf{R}) \mathcal{IG}_{\sigma_{ik}^2}(\nu_0, \gamma_0) \tag{10}$$

where $\mathbf{R}^{-1} = {\mathbf{Pa}_i^{(k)}}^\top \mathbf{Pa}_i^{(k)}$ and γ_0 and ν_0 take small positive real values. Based on these conjugate priors, the marginal likelihood can be obtained as

$$p(\mathbf{y}_i|\mathbf{Pa}_i^{(k)}) \propto |\mathbf{P}^\perp|^{\frac{1}{2}} (\gamma_0 + \mathbf{y}_i^\top \mathbf{P}^\perp \mathbf{y}_i)^{-\frac{N+\nu}{2}} \tag{11}$$

where $\mathbf{P}^\perp = \mathbf{I}_N - \mathbf{Pa}_i^{(k)} ({\mathbf{Pa}_i^{(k)}}^\top \mathbf{Pa}_i^{(k)} + \mathbf{R}^{-1})^{-1} {\mathbf{Pa}_i^{(k)}}^\top$ and $\mathbf{I}_N$ is an $N \times N$ identity matrix.

B. The topology prior $p(M_i^{(k)})$

We assume that each gene has the same *a priori* probability, say q, to be a parent gene. Thus, a Geometric distribution on the prior is expressed as

$$p(M^{(k)}) = q^{P_k} (1-q)^{G-P_k}. \tag{12}$$

where P_k denotes the total number of the parents under $M_i^{(k)}$ and G is the gross number of the gene. As a result, the number of the parents Q follows a Binomial distribution

$$p(Q = P_k) = \binom{G}{P_k} q^{P_k} (1-q)^{G-P_k}. \quad (13)$$

Since the mean number of parents $\bar{Q} = Gq$, the probability q can be calculated from the mean as

$$q = \bar{Q}/G. \quad (14)$$

Therefore, the choice of q reflects our prior knowledge about the average number of the parents.

The difficulties of the Bayesian structure learning are in two folds. First, the sample size N is normally much smaller than the total number of testing genes G. Secondly, the optimization and calculation of APPs themselves are NP hard and exact solutions are infeasible for large G.

C. The proposed solutions

To the end of first difficulty, we impose an upper limit Q_{max} on the number of parents and restrict $Q_{max} < N$. This constraint essentially forces us to search only among the topologies whose regulatory models are over-determined. It also serves to reduce the size of the search space and helps alleviate the second difficulty. To search the network topology from data, We proposed a reversible jump Markov chain Monte Carlo (RJMCMC) to approximate the MAP solution and the APPs. RJMCMC, proposed by Green in [7], is an MCMC algorithm for sampling from a joint topology-parameter space. In our case, the objective of the RJMCMC is to generate random samples from the APPs $p(M^{(k)}|\mathbf{Y})$. Then, the MAP solution can be approximated with the most-frequently-occurred samples. These samples can be also used to produce an approximate to the desired APPs.

The algorithm of the proposed RJMCMC is summarized in the following box.

Algorithm: RJMCMC

Provide an initial topology and assign it to $M(0)$. Iterate T times and at the tth iteration perform the following steps .

1) **Candidate selection**: Suppose $M(t-1) = M_i^{(k)}$. If $P_k = 1$, randomly select a gene from the non-parent genes; If $P_k = Q_{max}$, randomly select a gene from the parent genes; Otherwise, randomly select a gene from all G genes
2) If the gene is a parent in $M(t-1)$
   - **Death move**: Remove the node associated with the selected gene from $M^{(k)}$ to obtain topology $M_i^{(j)}$. Set $M(t) = M_i^{(j)}$ with probability $\lambda = \min\{BF(j,k), \alpha(j,k)\}/\alpha(j,k)$ Otherwise $M(t) = M_i^{(k)}$. else
   - **Birth move**: Add the node associated with the select gene to $M^{(k)}$ to obtain topology $M_i^{(l)}$. Set $M(t) = M_i^{(l)}$ with probability $\lambda = \min\{BF(l,k), \alpha(l,k)\}/\alpha(l,k)$. Otherwise $M(t) = M_i^{(k)}$.

In this algorithm, $BF(M_i, M_k)$ is the Bayes factor between M_i and M_k and is defined as

$$BF(j,k) = \frac{p(\mathbf{y}|\mathbf{Pa}_i^{(j)})}{p(\mathbf{y}|\mathbf{Pa}_i^{(k)})} \quad (15)$$

In addition, $\alpha(j,k)$ is calculated as the product of the topology prior ratio r_t and the probability ratio of moves r_m, i.e.,

$$\alpha(j,k) = r_t(j,k) r_m(j,k) \quad (16)$$

where

$$\begin{aligned} r_t(j,k) &= \frac{p(M_j)}{p(M_k)} \\ &= \begin{cases} \frac{1-q}{q} & \text{for death move} \\ \frac{q}{1-q} & \text{for birth move} \end{cases} \end{aligned} \quad (17)$$

and

$$r_m(j,k) = \begin{cases} \frac{Q_{max}}{G} & \text{if } P_k = Q_{max} \\ \frac{G-1}{G} & \text{if } P_k = 1 \\ 1 & \text{otherwise} \end{cases} \quad (18)$$

α can be considered as a threshold on Bayes factor BF. When $BF > \alpha$, the proposed move is accepted with probability of 1 and otherwise it is accepted with probability BF/α. This stochastic move can avoid being trapped on local high density regions and thus possibly produce a global solution.

When the algorithm finishes, there will be T samples of $M_i^{(k)}$ and we discard the first couple of samples (which are called burn-in) to account for convergence of Markov chain. Afterwards, supposing that there are T' samples left, the APPs can be approximated by

$$p(M_i^{(k)}) = \frac{1}{T'} \sum_{t=1}^{T'} \delta(M_i^{(k)} - M(t)) \quad (19)$$

where $\delta(\cdot)$ is the Kronecker Delta function and $M(t)$ denotes the tth sample in the final collection.

D. Parameter learning

Once we determine the topology of the network, the model parameters $\mathbf{w}_i$ and σ_i^2 can be estimated according to the minimum mean squared error (MMSE) criterion

$$\mathbf{w}_{i,MMSE} = \mathbf{B}^{-1} {\mathbf{Pa}_i^{(k)}}^\top \mathbf{y}_i \quad (20)$$

and

$$\sigma^2_{i,MMSE} = \frac{\frac{\mathbf{y}_i^\top \mathbf{P}^\perp \mathbf{y}_i + \gamma_0}{2}}{\frac{N+\nu_0}{2} - 1} \quad (21)$$

where $\mathbf{B} = {\mathbf{Pa}_i^{(k)}}^\top \mathbf{Pa}_i^{(k)} + \mathbf{R}^{-1}$, and $\mathbf{P}^\perp = \mathbf{I}_N - \mathbf{Pa}_i^{(k)} \mathbf{B}^{-1} {\mathbf{Pa}_i^{(k)}}^\top$.

IV. Test Results and Conclusion

We tested the proposed DBN and the RJMCMC learning algorithm on the cDNA microarray data of 58 genes in the yeast cell cycles, reported in [8]. The dataset contains 18 samples evenly measured over a period of 119 minutes. Missing values exist in this data sets and simple spline interpolation was used to fill in the missing data. When implementing the RJMCMC algorithm, we used $\gamma_0 = 0.36$ and $\nu_0 = 1.2$ and we set $T = 10,000$ and ran the algorithm 10 times independently. Also we set $Q_{max} = 5$ and assumed that on average there were 2 parents for each gene,which implies $q = 2/58$.

The inferred network is shown in Figure 2. In this network, the nodes are labeled with gene names. The thickness of the arrow is determined by the magnitude of the corresponding weight, which denotes the degree of regulation. If the weight is positive, up regulation would be implied and a solid edge is used for the arrow. Otherwise, a dash line is used, which represents down regulation. We compared the network with the KEGG pathway map (http://www.genome.jp/kegg/) and marked the unconfirmed regulations by blue edges. A confirmed regulation suggests a true positive in our inference results. The brown-shaded nodes are the genes that were not included in the KEGG map. We observed some general interaction networks supported by previous experimental and computational studies. However, many interactions appeared inconsistent with the current biological views presented in the KEGG map. We calculated the posterior distribution of the topology for each gene. We plot a the APPs of the topology of gene CDC28 in Figure 3. The largest and the second largest probabilities are 0.036 and 0.021. Because it is small, we do not have good confidence about this MAP solution. We also calculated the respective averages over the largest and the second largest *a posteriori* probabilities of all the genes and they are 0.0257 and 0.0203. We observed that the difference between the two probabilities is slim. This suggests that, in addition to the inferred network, there were competing topologies that are almost equally likely to be a solution.

In this paper, we proposed a dynamic Bayesian network modeling of time series microarray data. A RJMCMC algorithm is adopted for determining the network topology. The developed full Bayesian solution can provide information on the APPs of topology, which can be used as an indication to the confidence on the inferred results. We tested the proposed method on yeast microarray data in cell cycles. The estimated APPs indicated generally low confidence in the results. This is mainly due to the small data size and possibly inaccuracy in the assumed linear regulatory models.

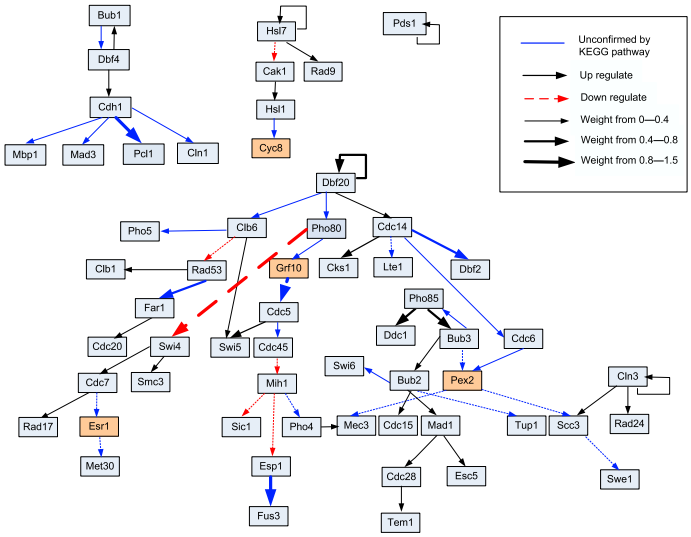

Fig. 2. The inferred gene network for $Q_{max} = 5$ and $q = 2/58$.

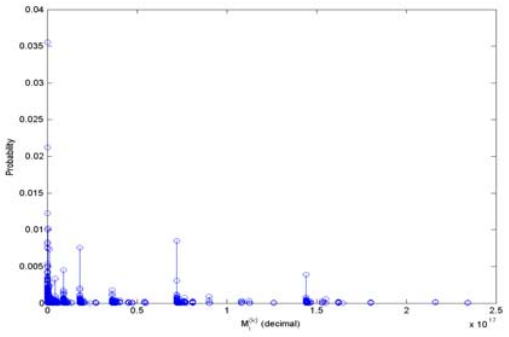

Fig. 3. The estimated posterior distribution of the topology for gene CDC28 in experiment 2. The x-axis is the decimal representation of $M_i^{(k)}$.

References

[1] P. Ḋhaeseleer, P. Liang, S. Fuhrman, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[2] S. Y. Kim, S. Imoto, and S. Miyano, "Inferring gene networks from time series microarray data using dynamic Bayesian networks," *Briefings in Bioinformatics*, vol. 4, no. 3, pp. 228–235, 2003.

[3] M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild, "A Beyesian approach to reconstructing genetic regulatory networks with hidden factors," *Bioinformatics*, vol. 20, pp. 1361–1372, Sept. 2004.

[4] C. Rangel, D. L.Wild, F. Falciani, Z. Ghahramani, and A. Gaiba, "Modeling biological responses using gene expression profiling and linear dynamical systems," in *Proceedings of the 2nd International Conference on Systems Biology*, 2001.

[5] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1997.

[6] J. M. Bernardo and A. F. Smith, Eds., *Bayesian Theory*. John Wiley and Son Ltd, 2000.

[7] P. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," vol. 82, pp. 711–732, 1995.

[8] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D.Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, pp. 3273–3297, 1998.