

ON CONSTRUCTION OF STOCHASTIC GENETIC NETWORKS BASED ON GENE EXPRESSION SEQUENCES

WAI-KI CHING

*Department of Mathematics, The University of Hong Kong
Pokfulam Road, Hong Kong, Hong Kong, China
wkc@maths.hku.hk*

MICHAEL M. NG

*Department of Mathematics, Hong Kong Baptist University
Kowloon Tong, Hong Kong, Hong Kong, China
mng@math.hkbu.edu.hk*

ERIC S. FUNG

*Department of Mathematics, The University of Hong Kong
Pokfulam Road, Hong Kong, Hong Kong, China
ericfung@graduate.hku.hk*

TATSUYA AKUTSU

*Institute for Chemical Research, Kyoto University
Gokasho Uji, Kyoto 611-0011, Japan, Kyoto, Japan
takutsu@kuicr.kyoto-u.ac.jp*

Reconstruction of genetic regulatory networks from time series data of gene expression patterns is an important research topic in bioinformatics. Probabilistic Boolean Networks (PBNs) have been proposed as an effective model for gene regulatory networks. PBNs are able to cope with uncertainty, corporate rule-based dependencies between genes and discover the sensitivity of genes in their interactions with other genes. However, PBNs are unlikely to use directly in practice because of huge amount of computational cost for obtaining predictors and their corresponding probabilities. In this paper, we propose a multivariate Markov model for approximating PBNs and describing the dynamics of a genetic network for gene expression sequences. The main contribution of the new model is to preserve the strength of PBNs and reduce the complexity of the networks. The number of parameters of our proposed model is $O(n^2)$ where n is the number of genes involved. We also develop efficient estimation methods for solving the model parameters. Numerical examples on synthetic data sets and practical yeast data sequences are given to demonstrate the effectiveness of the proposed model.

Keywords: Genetic networks; probabilistic Boolean networks; multivariate Markov chains; gene expression sequences.

1. Introduction

One of important research topics in genomics and systems biology is to understand the mechanism in which cells execute and control a huge number of operations for normal functions, and also the way in which the cellular systems fail in diseases. Various kinds of models have been proposed for solving this problem, based on such methods as neural networks, non-linear ordinary differential equations,

Petri nets, see for instance Smolden *et al.*,²⁹ Bower² and de Jong.⁹ Another approach is to model the genetic regulatory system by a Boolean network and infer the network structure and parameters by real gene expression data. The strengths and weaknesses of different genetic modeling approaches have been analyzed in Wessels *et al.*³² Then by using the inferred network model, one may be able to discover the underlying gene regulatory mechanisms

and therefore it helps to make useful predictions by using computer simulations. The Boolean network model was first introduced by Kauffman.¹⁸ Good views of this model can be found in Akutsu *et al.*,¹ Kauffman,¹⁸ and Shmulevich *et al.*^{26–28} In this network model, each gene is regarded as a vertex of the network and is quantized into two levels only (express (0) or not-express (1)). Akutsu *et al.*¹ proposed the noisy Boolean networks together with an identification algorithm. In their model, they relax the requirement of consistency imposed by the Boolean functions. Regarding the effectiveness of a Boolean formalism, Shmulevich *et al.*^{26–28} proposed a Probabilistic Boolean network (PBN) that can share the appealing rule-based properties of Boolean networks and it is robust in the presence of uncertainty. Their model is able to show a clear separation between different subtypes of gliomas as well as between different sarcomas by using multi-dimensional scaling. A logical representation of cell cycle regulation can also be found in Shmulevich *et al.*^{26–28}. Furthermore, a structural intervention method is proposed for controlling the stationary behavior in PBNs, see Shmulevich *et al.*^{26–28} It should be noted that many of existing studies on genetic network reconstruction employ gene expression data obtained by using the DNA microarray technology, which is based on hybridization of cDNAs and differential expression. However, it is widely recognized that for DNA microarrays, there exist several problems on reproducibility of measurements and between-slide variation, see for instance Chen *et al.*,⁵ Khan *et al.*¹⁹ and Tsodikov *et al.*³⁰ Moreover, genetic regulation also exhibits uncertainty on the biological level. Therefore, it seems reasonable to use probabilistic models such as PBNs for genetic network reconstruction from microarray data.

The dynamics of a PBN can be studied in the context of a standard Markov chain. However, for both noisy Boolean networks and the PBNs, the number of parameters (Boolean functions) grows exponentially with respect to the number of genes n , and therefore heuristic methods are needed for model training Akutsu *et al.*¹ The main contribution of this paper is to propose a multivariate Markov model (a stochastic model) which allows both the intra- and inter-transition probabilities among the gene expression sequences. The number of parameters in the model

is only $O(n^2)$. We develop efficient model parameters estimation methods based on linear programming. We also propose a method for recovering the structure and rules for a given Boolean network.

The rest of the paper is organized as follows. In Sec. 2, we give a review on both Boolean networks and Probabilistic Boolean networks. In Sec. 3, we propose the multivariate Markov model for Boolean networks. In Sec. 4, the estimation methods for the probability of a predictor function based on our model parameters are illustrated. In Sec. 5, the estimation methods for model parameters are given. In Sec. 6, we apply the proposed model and method to both synthetic data and practical gene expression data of yeast. Finally, concluding remarks are given to address further research issues in Sec. 7.

2. Probabilistic Boolean Network

Boolean network models are commonly used for studying generic coarse-grained properties of large genetic networks without knowing specific quantitative details. Boolean network is deterministic, the only uncertainty can be introduced is the initial starting state. Generally speaking, a Boolean network $G(V, \mathcal{F})$ consists of a set of nodes

$$V = \{v_1, v_2, \dots, v_n\}$$

and $v_i(t)$ represents the state (0 or 1) of v_i at time t . A list of Boolean functions

$$\mathcal{F} = \{f^{(1)}, f^{(2)}, \dots, f^{(n)}\}$$

represents the rules regulatory interaction between nodes:

$$v_i(t+1) = f^{(i)}(\mathbf{v}(t)), \quad i = 1, 2, \dots, n,$$

where

$$\mathbf{v}(t) = (v_1(t), v_2(t), \dots, v_n(t)).$$

In general, there may exist some unnecessary nodes in a Boolean function. For a Boolean function $f^{(j)}$, the variable $v_i(t)$ is fictitious if

$$\begin{aligned} & f^{(j)}(v_1(t), \dots, v_{i-1}(t), 0, v_{i+1}(t), \dots, v_n(t)) \\ &= f^{(j)}(v_1(t), \dots, v_{i-1}(t), 1, v_{i+1}(t), \dots, v_n(t)) \end{aligned}$$

for all possible values of

$$v_1(t), \dots, v_{i-1}(t), v_{i+1}(t), \dots, v_n(t).$$

We remark that when a Boolean network is used in the construction of underlying genetic networks,

then n represents the number of genes under consideration, each vertex v_i represents the i th gene, and $v_i(t)$ represents the expression level of the i th gene at time t , taking either 0 or 1. The expression level of each gene is functionally related to that of other genes. Computational models that reveal these logical relations have been constructed in Bodnar *et al.*,⁴ Mendoza *et al.*²² and Huang *et al.*¹⁵

Standard Boolean networks are deterministic. However, assuming an inherent determinism is not quite reasonable as it assumes a biological environment having no uncertainty. The existence of regularity of genetic function and interaction is caused by intrinsic self-organizing stability of the dynamical system instead of “hard-wired” logical rules, Shmulevich *et al.*^{26–28}. In the empirical aspect, sample noise and relatively small amount of samples may cause incorrect results in logical rules. In order to overcome the deterministic rigidity of Boolean networks, the development of Probabilistic Boolean networks (PBNs) is essential. Not only PBN shares the appealing properties of Boolean networks, but also it is able to cope with the uncertainty, including the data and model selection, Shmulevich *et al.*^{26–28}

PBNs were firstly proposed by Shmulevich *et al.*^{26–28} for genetic regulatory network. The model can be written as:

$$\mathcal{F}_i = \{f_j^{(i)}\}_{j=1, \dots, l(i)},$$

where each $f_j^{(i)}$ is a predictor determining the value of the gene v_i and $l(i)$ is the number of possible predictors for the gene v_i . It is clear that

$$\mathcal{F} = \bigcup_{i=1}^n \mathcal{F}_i.$$

We notice that when the number of possible PBN realization N is equal to 1 (i.e., $\prod_{i=1}^n l(i) = 1$), the PBN reduces to the standard Boolean network. Let $c_j^{(i)}$ be the probability that the j th predictor, $f_j^{(i)}$, is chosen to predict a state of the i th gene and this probability can be estimated by Coefficient of Determination (COD); Dougherty *et al.*¹⁰ Let us briefly describe COD here. Let $\epsilon_j^{(i)}$ be the optimal error achieved by $f_j^{(i)}$ and ϵ_i is the error of best estimate of i th gene in the absence of any conditional variable, then we have

$$\theta_j^{(i)} = \frac{\epsilon_i - \epsilon_j^{(i)}}{\epsilon_i}.$$

For all positive $\theta_j^{(i)}$, we can obtain $c_j^{(i)}$ by:

$$c_j^{(i)} = \begin{cases} \frac{\theta_j^{(i)}}{\sum_{k=1}^{l(i)} \{\theta_k^{(i)} : \theta_k^{(i)} > 0\}}, & \text{if } \theta_j^{(i)} \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

and $c_j^{(i)}$ must satisfies

$$\sum_{j=1}^{l(i)} c_j^{(i)} = 1. \quad \text{for } i = 1, \dots, n.$$

For any given time point, the expression level of the i th gene is determined by one of the possible predictors $f_j^{(i)}$ for $1 \leq j \leq l(i)$. The probability of a transition from $\mathbf{v}(t)$ to $\mathbf{v}(t+1)$ can be obtained as follows:

$$\prod_{i=1}^n \left[\sum_{k=1}^{l(i)} \{c_k^{(i)} : f_k^{(i)}(\mathbf{v}(t)) = v_i(t+1)\} \right].$$

On the other hand, the level of influences from gene j to gene i can be estimated by $I_j(v_i) =$

$$\sum_{k=1}^{l(i)} \text{Prob}(f_k^{(i)}(v_1, \dots, v_{j-1}, 0, v_{j+1}, \dots, v_n) \neq f_k^{(i)}(v_1, \dots, v_{j-1}, 1, v_{j+1}, \dots, v_n)) c_k^{(i)}. \quad (1)$$

Before evaluating either state transition probabilities or $I_j(v_i)$, we first need to obtain all the predictors $\bigcup_{i=1}^n \mathcal{F}_i$. We remark that for each set of \mathcal{F}_i with $1 \leq i \leq n$, the maximum number of possible predictors is equal to 2^{2^n} as $1 \leq l(i) \leq 2^{2^n}$. This is also true for their corresponding probabilities $\{c_1^{(i)}, \dots, c_{l(i)}^{(i)}\}$. It implies that the number of parameters in the PBN model is of $O(n2^{2^n})$. Obviously, the number of parameters increases double-exponentially with respect the number of genes n . Moreover, the COD used in obtaining $c_k^{(i)}$ must be estimated from the training data. Hence, it is almost impractical to apply this model due to either its model complexity or parameters imprecision owing to limited sample size. For the microarray-based analysis done by Kim *et al.*,²⁰ the number of genes in each set of \mathcal{F}_i was kept to a maximum of three.

We notice that PBN is a discrete-time process, the probability distribution of gene expression at time $t+1$ of the i th gene can be estimated by the gene expression of other n genes at time t via one-lag transition matrix. This is a Markov process framework. In next section, we propose and develop a

multivariate Markov model for the construction of genetic networks.

3. Multivariate Markov Chain Models

In this section, we propose our multivariate Markov chain model to infer the genetic network of n genes. In our model, each gene sequence can be regarded as a Markov chain with two states (0 or 1) and no prior information on n genes relationships is assumed. Our proposed model is used to uncover the underlying various gene relationships, including genes and genes cyclic or acyclic relationships.

In a standard PBN network $G(V, \mathcal{F})$, a single sequence (corresponding to time series data for one gene) in PBNs $\{v(1), \dots, v(T)\}$ is logically represented as a sequence of vectors V_1, \dots, V_T , where T is the length of the sequence, and $V_j = E_k$ (E_k is the unit column vector with the $(k+1)$ th entry being one) if $v(j) = k$ (i.e., it is in the expression level k , $k = 0$ or 1). A first-order discrete-time Markov chain satisfies the following relationship:

$$\begin{aligned} \text{Prob}(V_{t+1} = E_{v(t+1)} \mid V_0 = E_{v(0)}, \dots, V_t = E_{v(t)}) \\ = \text{Prob}(V_{t+1} = E_{v(t+1)} \mid V_t = E_{v(t)}). \end{aligned}$$

These probabilities are assumed to be independent of t and can be written as

$$p_{ij} = \text{Prob}(V_{t+1} = E_i \mid V_t = E_j), \quad \forall i, j \in \{0, 1\}.$$

The matrix P , such that $P_{ij} = p_{ij}$ is called the transition probability matrix.

For a network of n genes, our model assumes the following relationship among the gene expression sequences:

$$\mathbf{V}_{t+1}^{(j)} = \sum_{k=1}^n \lambda_{jk} P^{(jk)} \mathbf{V}_t^{(k)}, \quad j = 1, 2, \dots, n \quad (2)$$

where

$$\lambda_{jk} \geq 0 \quad \text{for } 1 \leq j, k \leq n \quad \text{and} \quad \sum_{k=1}^n \lambda_{jk} = 1, \quad (3)$$

and $\mathbf{V}_t^{(i)}$ is the expression level probability distribution of the i th gene at the time t . From the above relationship, the expression probability distribution of the j th gene at the $(t+1)$ th base depends on the weighted average of $P^{(jk)} \mathbf{V}_t^{(k)}$. The most proper parent genes for j th gene (i.e., $\mathbf{V}_{t+1}^{(j)}$) can be retrieved from the corresponding λ_{jk} . The higher value of λ_{jk} implies stronger parents and descendants relationship between j th and k th gene. When this process is

repeated for each j , the whole genetic network could be constructed. If there exists a set of genes

$$\{V^{(j_h)} : h = 1, 2, \dots, w \text{ and } j_h \in (1, 2, \dots, n)\}$$

such that for any gene in this set, the rest of genes are the only candidate being a corresponding parent gene, then this set of genes forms a cycle. Here $P^{(jk)}$ is a transition probability matrix from the k th gene to the j th gene. In matrix form, we write

$$\mathbf{V}_{t+1} \equiv \begin{pmatrix} \mathbf{V}_{t+1}^{(1)} \\ \mathbf{V}_{t+1}^{(2)} \\ \vdots \\ \mathbf{V}_{t+1}^{(n)} \end{pmatrix} = Q \begin{pmatrix} \mathbf{V}_t^{(1)} \\ \mathbf{V}_t^{(2)} \\ \vdots \\ \mathbf{V}_t^{(n)} \end{pmatrix} \equiv Q \mathbf{V}_t$$

where

$$Q = \begin{pmatrix} \lambda_{11}P^{(11)} & \lambda_{12}P^{(12)} & \dots & \lambda_{1n}P^{(1n)} \\ \lambda_{21}P^{(21)} & \lambda_{22}P^{(22)} & \dots & \lambda_{2n}P^{(2n)} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{n1}P^{(n1)} & \lambda_{n2}P^{(n2)} & \dots & \lambda_{nn}P^{(nn)} \end{pmatrix}.$$

Each column sum of Q is not necessarily equal to one and therefore it is not a transition probability matrix. But the column sum of each of $P^{(jk)}$ is equal to one, so it is a transition probability matrix. In fact, the one-step transition probability matrices $P^{(jk)}$ describe the transition of states from sequence j to sequence i .

Here we present two theorems on our multivariate model as in Ching *et al.*⁶ Before that, we need the following two lemmas. The proofs of the lemmas can be found in Berman *et al.*³ Given a transition probability matrix P of a finite Markov chain, we have

Lemma 1. *The matrix P has an eigenvalue equal to one and all the eigenvalues of P must have modulus less than or equal to one.*

Lemma 2 (Perron-Frobenius Theorem). *Let A be a nonnegative and irreducible square matrix of order m . Then we have*

- (i) *A has a positive real eigenvalue λ which is equal to its spectral radius, i.e.,*

$$\lambda = \max_k |\lambda_k(A)|$$

where $\lambda_k(A)$ denotes the k th eigenvalue of A .

- (ii) There corresponds an eigenvector \mathbf{z} with all its entries being real and positive, such that $A\mathbf{z} = \lambda\mathbf{z}$.
- (iii) λ is a simple eigenvalue of A .

Theorem 1. If $[\lambda_{jk}]$ is irreducible for $1 \leq j, k \leq n$, then the matrix Q has an eigenvalue equal to 1 and the eigenvalues of Q have modulus less than or equal to 1.

Proof. By using (3), the column sum of the following matrix

$$\Lambda = \begin{pmatrix} \lambda_{1,1} & \lambda_{2,1} & \cdots & \lambda_{s,1} \\ \lambda_{1,2} & \lambda_{2,2} & \cdots & \lambda_{s,2} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{1,s} & \lambda_{2,s} & \cdots & \lambda_{s,s} \end{pmatrix}$$

is equal one. Since Λ is nonnegative and irreducible, by Lemma 2 (Perron-Frobenius Theorem) there exists a vector

$$\mathbf{y} = [y_1, y_2, \dots, y_s]^T$$

such that

$$\Lambda\mathbf{y} = \mathbf{y} \quad \text{or} \quad \mathbf{y}^T \Lambda^T = \mathbf{y}^T.$$

We note that

$$\mathbf{1}_2 P^{(ij)} = \mathbf{1}_2, \quad 1 \leq i, j \leq s,$$

where $\mathbf{1}_2$ is the $1 \times m$ vector of all ones, i.e.,

$$\mathbf{1}_2 = [1, 1, \dots, 1].$$

Then it is easy to show that

$$[y_1 \mathbf{1}_2, y_2 \mathbf{1}_2, \dots, y_s \mathbf{1}_2] Q = [y_1 \mathbf{1}_2, y_2 \mathbf{1}_2, \dots, y_s \mathbf{1}_2].$$

and hence one is an eigenvalue of Q .

Next we show that all the eigenvalues of Q are less than or equal to one. Let us define

$$\begin{aligned} \|\mathbf{z}\| &\equiv \max_{1 \leq i \leq s} \{\|\mathbf{z}_i\|_1 : \mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_s], \\ &\quad \mathbf{z}_j \in \mathbf{R}^2, 1 \leq j \leq s\}. \end{aligned}$$

It is straightforward to show that $\|\cdot\|$ is a norm on \mathbf{R}^{2s} . It follows that we have the following matrix norm

$$\|Q\| \equiv \sup\{\|Q\mathbf{z}\| : \|\mathbf{z}\| = 1\}.$$

Since $P^{(ij)}$ is a transition matrix, each element of $P^{(ij)}$ are less than or equal to 1. We have

$$\|P^{(ij)}\mathbf{z}_j\|_1 \leq \|\mathbf{z}_j\|_1 \leq 1, \quad 1 \leq i, j \leq s.$$

It follows that

$$\begin{aligned} &\|\lambda_{i1} P^{(i1)} \mathbf{z}_1 + \lambda_{i2} P^{(i2)} \mathbf{z}_2 + \cdots + \lambda_{is} P^{(is)} \mathbf{z}_s\|_1 \\ &\leq \|\mathbf{z}\| \cdot \sum_{j=1}^s \lambda_{ij} = 1, \end{aligned}$$

for $1 \leq i \leq s$ and hence $\|Q\| \leq 1$. Since the spectral radius of Q is always less than or equal to any matrix norm of Q and the result follows. \square

Theorem 2. Suppose that $P^{(jk)}$ ($1 \leq j, k \leq n$) are irreducible and $\lambda_{jk} > 0$ for $1 \leq j, k \leq n$. Then there is a vector

$$\bar{\mathbf{V}} = [\bar{\mathbf{V}}^{(1)}, \bar{\mathbf{V}}^{(2)}, \dots, \bar{\mathbf{V}}^{(n)}]^T$$

such that $\bar{\mathbf{V}} = Q\bar{\mathbf{V}}$ and

$$\sum_{i=1}^m [\bar{\mathbf{V}}^{(j)}]_i = 1, \quad 1 \leq j \leq n.$$

where m is the number of states.

Proof. By Lemmas 1 and 2, there is exactly one eigenvalue of Q equal to one. This implies that

$$\lim_{n \rightarrow \infty} Q^n = \mathbf{v}\mathbf{u}^T$$

is a positive rank one matrix as Q is irreducible. Therefore

$$\lim_{n \rightarrow \infty} \mathbf{V}_{n+1} = \lim_{n \rightarrow \infty} Q^n \mathbf{V}_0 = \mathbf{v}\mathbf{u}^T \mathbf{V}_0 = \alpha\mathbf{v}.$$

Here α is a positive number since $\mathbf{V} \neq 0$ and is non-negative. This implies that \mathbf{V}_n tends to a stationary vector as n goes to infinity. Finally, we note that if \mathbf{V}_0 is a vector such that

$$\sum_{i=1}^m [\mathbf{V}_0^{(j)}]_i = 1, \quad 1 \leq j \leq s,$$

then $Q\mathbf{V}_0$ is also a vector having this property. Hence the result follows. We also note that according to this theorem, the vector \mathbf{V} is not a probability distribution vector, but $\mathbf{V}^{(j)}$ is a probability distribution vector. \square

4. Efficient Estimation of the Probability of Predictor Function $c_g^{(d)}$

From our own model parameters, it is sufficient to uncover the gene regulatory network. However,

one would like to have a performance comparison between PBNs and our model. Thus, we illustrate a method to estimate commonly used PBN parameters efficiently from our model parameters. In PBNs with n genes, there are n disjoint sets of predictors \mathcal{F}_i and each of them is used for an unique gene sequence. In particular, for the d th set of predictors \mathcal{F}_d , we notice that the possibility corresponding to each predictor $f_j^{(d)}$ can be obtained from our probability stationary vector and the detail is given as follows. First, by replacing j by d in Eq. (2), one can estimate the conditional probability distribution $X_{i_1, \dots, i_n}^{(d)}$ for d output expression at base $t+1$ given by a set of genes input expression at base t , i.e.,

$$\begin{aligned} X_{i_1, \dots, i_n}^{(d)} &= \text{Prob}(V_{t+1}^{(d)} \mid V_t^{(k)} = E_{i_k} \text{ for } k = 1, \dots, n) \\ &= \sum_{k=1}^n \lambda_{dk} P^{(dk)} E_{i_k} \\ &= \sum_{k=1}^n \lambda_{dk} P_{(\cdot, i_k)}^{(dk)} \end{aligned}$$

where $i_k \in \{0, 1\}$ and $P_{(\cdot, i)}^{(dk)}$ denote the i column of $P^{(dk)}$. Clearly, each probability vector $X_{i_1, \dots, i_n}^{(d)}$ is a unit vector and for each d , there are 2^n number of probability vectors we need to estimate. If $\lambda_{dj}=0$ for some $j \in \{1, \dots, n\}$, it represents that the j th gene does not have any influence to the d th gene, and

$$X_{i_1, \dots, i_{j-1}, 0, i_{j+1}, \dots, i_n}^{(d)} \equiv X_{i_1, \dots, i_{j-1}, 1, i_{j+1}, \dots, i_n}^{(d)}$$

the number of estimated probability vectors could be reduced by half. After all the essential $X_{i_1, \dots, i_n}^{(d)}$ have been estimated, the probability $c_g^{(d)}$ of the predictor $f_g^{(d)}$ can be estimated by

$$c_g^{(d)} = \prod_{i_k \in \{0, 1\}, k=1, \dots, n} X_{i_1, \dots, i_n}^{(d)} (f_g^{(d)}(i_1, \dots, i_n) + 1)$$

where $X_{i_1, \dots, i_n}(h)$ denotes the h entry of the vector X_{i_1, \dots, i_n} and $f_g^{(d)}(i_1, \dots, i_n) \in \{0, 1\}$. If $c_g^{(d)} = 0$, then the predictor function $f_g^{(d)}$ does not exist and it should be eliminated. It is interesting to justify how the expression of i th gene is affected by the expression of j th gene, therefore, the degree of sensitivity from j th gene to i th gene can be estimated by Eq. (1) mentioned in previous section. We notice that there are two situations that $I_j(V_i) = 0$, Shmulevich *et al.* (2002), namely,

- If $\lambda_{ij} = 0$, then j th gene does not give any influence on i th gene.

- The first two columns of the matrix $P^{(ij)}$ are identical, that means no matter the expression of j th gene is, the result of the probability vector is not affected.

5. Model Parameters Estimation

In this section, we propose methods to estimate $P^{(jk)}$ and λ_{jk} . For the i th and the j th of the n genes, we estimate the transition probability matrix $P^{(jk)}$ by the following method. We first count the transition frequency from one expression in j th gene to another expression in i th gene. After the usual normalization, we obtain an estimate of the transition probability matrix. Totally, we have to estimate n^2 m -by- m such transition probability matrices for the multivariate Markov model. Here m is the number of possible expression levels and can take any positive integers. In our application here, we take $m = 2$. More precisely, we count the transition frequency $d_{i_j i_k}^{(jk)}$ from the expression i_k in the gene $\{v_k\}$ at time t to the expression i_j in the gene $\{v_j\}$ at time $t+1$ and therefore we construct the transition frequency matrix for the sequences as follows:

$$D^{(jk)} = \begin{pmatrix} d_{11}^{(jk)} & \cdots & d_{1m}^{(jk)} \\ \vdots & \ddots & \vdots \\ d_{m1}^{(jk)} & \cdots & d_{mm}^{(jk)} \end{pmatrix}.$$

From $D^{(jk)}$, we obtain the estimate for $P^{(jk)}$ as follows:

$$\hat{P}^{(jk)} = \begin{pmatrix} \hat{p}_{11}^{(jk)} & \cdots & \hat{p}_{1m}^{(jk)} \\ \vdots & \ddots & \vdots \\ \hat{p}_{m1}^{(jk)} & \cdots & \hat{p}_{mm}^{(jk)} \end{pmatrix}$$

where

$$\hat{p}_{ab}^{(jk)} = \begin{cases} \frac{d_{ab}^{(jk)}}{\sum_{a=1}^m d_{ab}^{(jk)}} & \text{if } \sum_{a=1}^m d_{ab}^{(jk)} \neq 0 \\ \frac{1}{m} & \text{otherwise.} \end{cases}$$

Though the estimation method may be modified for using pseudocounts, some procedure will be required for adjusting pseudocount parameters.

Besides $\hat{P}^{(jk)}$, we need to estimate the parameters λ_{jk} . We have shown that the multivariate Markov model has a “stationary vector of distributions” $\bar{\mathbf{V}}$ in Theorem 2. The vector $\bar{\mathbf{V}}$ can be

estimated from the gene expression sequences by computing the proportion of the occurrence of each expression gene and let us denote it by

$$\hat{\mathbf{V}} = (\hat{\mathbf{V}}^{(1)}, \hat{\mathbf{V}}^{(2)}, \dots, \hat{\mathbf{V}}^{(n)})^T.$$

We therefore expect $\hat{Q}\hat{\mathbf{V}} \approx \hat{\mathbf{V}}$ where

$$\hat{Q} = \begin{pmatrix} \lambda_{11}\hat{P}^{(11)} & \lambda_{12}\hat{P}^{(12)} & \dots & \lambda_{1n}\hat{P}^{(1n)} \\ \lambda_{21}\hat{P}^{(21)} & \lambda_{22}\hat{P}^{(22)} & \dots & \lambda_{2n}\hat{P}^{(2n)} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{n1}\hat{P}^{(n1)} & \lambda_{n2}\hat{P}^{(n2)} & \dots & \lambda_{nn}\hat{P}^{(nn)} \end{pmatrix}.$$

From above equation, it suggests one possible way to estimate the parameters $\Lambda = \{\lambda_{jk}\}$ as follows. For each j , we consider the following optimization problem:

$$\min_{\lambda} \|\hat{Q}\hat{\mathbf{V}} - \hat{\mathbf{V}}\|_{\infty}$$

subject to

$$\sum_{k=1}^n \lambda_{jk} = 1, \quad \text{and} \quad \lambda_{jk} \geq 0, \quad \forall k$$

or equivalently

$$\min_{\lambda} \max_i \left\| \left[\sum_{k=1}^n \lambda_{jk} \hat{P}^{(jk)} \hat{\mathbf{V}}^{(k)} - \hat{\mathbf{V}}^{(j)} \right]_i \right\|$$

subject to

$$\sum_{k=1}^n \lambda_{jk} = 1, \quad \text{and} \quad \lambda_{jk} \geq 0, \quad \forall k.$$

Here $[\cdot]_i$ denote the i th entry of the vector. This optimization problem is equivalent to the following linear program:

$$\min_{\lambda} w_j$$

subject to

$$\begin{pmatrix} w_j \\ w_j \\ \vdots \\ w_j \end{pmatrix} \geq \hat{\mathbf{V}}^{(j)} - B \begin{pmatrix} \lambda_{j1} \\ \lambda_{j2} \\ \vdots \\ \lambda_{jn} \end{pmatrix}$$

and

$$\begin{pmatrix} w_j \\ w_j \\ \vdots \\ w_j \end{pmatrix} \geq -\hat{\mathbf{V}}^{(j)} + B \begin{pmatrix} \lambda_{j1} \\ \lambda_{j2} \\ \vdots \\ \lambda_{jn} \end{pmatrix},$$

for all $1 \leq j \leq n$

$$w_j \geq 0, \quad \sum_{k=1}^n \lambda_{jk} = 1, \quad \text{and} \quad \lambda_{jk} \geq 0, \quad \forall k.$$

where

$$B = [\hat{P}^{(j1)}\hat{\mathbf{V}}^{(1)} \mid \hat{P}^{(j2)}\hat{\mathbf{V}}^{(2)} \mid \dots \mid \hat{P}^{(jn)}\hat{\mathbf{V}}^{(n)}].$$

Since there are n independent linear programming (LP) problems of each being n constraints and n variables, the expected total computational complexity of solving such LPs is of $O(n^4)$, see for instance Fang and Puthenpura.¹¹ A worked example of the proposed algorithm can be found in Ching *et al.*⁷

5.1. The fitness of our model

Given all the state vectors $\mathbf{V}_t^{(k)}$ with $k = 1, \dots, n$, the state probability distribution $\mathbf{V}_{t+1}^{(k)}$ can be estimated by using (2). According to this state probability distribution, one may predict the state $\hat{\mathbf{v}}_j(t+1)$ for the j th sequence at time $t+1$ by taking the state with the maximum probability, i.e.,

$$\hat{\mathbf{v}}_j(t+1) = k - 1,$$

if $[\hat{\mathbf{V}}^{(j)}(t+1)]_i \leq [\hat{\mathbf{V}}^{(j)}(t+1)]_k$ for all $1 \leq i \leq 2$.

By making use of this treatment, our multivariate Markov model can be used to uncover the rules (build a truth table) for PBNs. With higher prediction accuracy, we have more confidence that the true genetic networks are uncovered by our model. To evaluate the performance and effectiveness, the prediction accuracy of all individual sequences r and the joint sequences R are defined respectively as follow:

$$r = \frac{1}{nT} \times \sum_{i=1}^n \sum_{t=1}^T \delta_t^{(i)} \times 100\%,$$

where

$$\delta_t^{(i)} = \begin{cases} 1, & \text{if } \hat{\mathbf{v}}_i(t) = \mathbf{v}_i(t) \\ 0, & \text{otherwise.} \end{cases}$$

and

$$R = \frac{1}{T} \times \sum_{t=1}^T \delta_t \times 100\%,$$

where

$$\delta_t = \begin{cases} 1, & \text{if } \hat{\mathbf{v}}_i(t) = \mathbf{v}_i(t) \text{ for all } 1 \leq i \leq n \\ 0, & \text{otherwise.} \end{cases}$$

Here T is the length of the data sequence. From the values of r and R , the accuracy of network realization for an individual sequence and for a whole set of sequences could be determined respectively.

6. Numerical Experiments

In this section, we test our multivariate Markov models with synthetic data sets and practical gene expression sequences of yeast with a PC having the following configurations: CPU = AMD 1800+, RAM = 512 Mb and OS = Windows 2000 Professional. Here, we present three synthetic data tests. In each of the tests, the experiments are repeated with five different scenarios, they are noise free (0.0%), 2.5% noise, 5% noise, 10% noise and 20% noise. For instance, in the case of 2.5% noise, we introduce random errors in the generated data sequences by randomly changing 2.5% of the data sequences. Furthermore, in solving the linear programming problem, both $\|\cdot\|_1$ and $\|\cdot\|_\infty$ were tried for all the numerical experiments. Since the numerical results of both norms are quite similar, we only report the result of using $\|\cdot\|_\infty$.

6.1. Test with synthetic data I

In the following, we give an example of a Boolean network having four vertices. The Boolean network is characterized by the truth table given in Table 1. Given an input (I_1, I_2, I_3, I_4) , the network has an output (O_1, O_2, O_3, O_4) . There are $2^4 = 16$ possible inputs (states) and their outputs (states) are listed in Table 1. We denote the states by S_i . With the structural rules, the system ultimately transits into so-called the attractor states Wuensche.³³ In fact, there are two cycles (two sets of attractor states) in this Boolean network:

$$S_7 \rightarrow S_{13} \rightarrow S_{14} \rightarrow S_{10} \rightarrow S_7 \quad \text{and} \quad S_2 \rightarrow S_2.$$

Let us denote the two cycles by C_1 and C_2 respectively. Clearly beginning with any initial state S_i , eventually the network state will enter into either the absorbing states form by C_1 or C_2 . Hence the 16 states of the Boolean network can be classified into two separate sets as follow:

$$D_1 = \{S_1, S_3, S_4, S_7, S_{10}, S_{13}, S_{14}, S_{16}\}$$

and

$$D_2 = \{S_2, S_5, S_6, S_8, S_9, S_{11}, S_{12}, S_{15}\}.$$

Here, we test our multivariate Markov model with synthetic data generated by the truth table in Table 1. The multivariate Markov model results are

Table 1. The truth table of the four-vertex Boolean network.

	Input				Output				
	I_1	I_2	I_3	I_4	O_1	O_2	O_3	O_4	
S_1	0	0	0	0	1	1	1	1	S_{16}
S_2	0	0	0	1	0	0	0	1	S_2
S_3	0	0	1	0	1	1	0	1	S_{13}
S_4	0	1	0	0	1	0	1	1	S_{14}
S_5	1	0	0	0	0	1	1	1	S_{15}
S_6	0	0	1	1	1	1	0	0	S_{11}
S_7	0	1	0	1	1	1	0	1	S_{13}
S_8	1	0	0	1	0	0	0	1	S_2
S_9	0	1	1	0	1	0	0	0	S_5
S_{10}	1	0	1	0	0	1	0	1	S_7
S_{11}	1	1	0	0	1	1	1	0	S_{12}
S_{12}	1	1	1	0	0	1	1	0	S_9
S_{13}	1	1	0	1	1	0	1	1	S_{14}
S_{14}	1	0	1	1	1	0	1	0	S_{10}
S_{15}	0	1	1	1	1	0	0	1	S_8
S_{16}	1	1	1	1	0	1	0	0	S_4

reported in Tables 2(a) and 2(b). We observe from Tables 2(a) and 2(b) that the prediction accuracy slightly decreases when the noise percentage increases linearly.

For each of the states $S_i (i = 1, 2, \dots, 16)$, we generate a sequence of $(v_1(t), v_2(t), v_3(t), v_4(t))$ of length

Table 2(a). Prediction accuracy of the multivariate Markov model (Part 1).

Noise	Prediction accuracy in %					
	0.0%		2.5%		5%	
	$r\%$	$R\%$	$r\%$	$R\%$	$r\%$	$R\%$
Initial state						
S_1	75	25	68	23	69	21
S_2	100	100	98	91	95	83
S_3	81	49	79	45	77	55
S_4	98	95	94	83	93	79
S_5	81	48	77	40	71	36
S_6	81	49	78	46	76	38
S_7	99	98	96	89	94	79
S_8	100	100	98	91	95	82
S_9	100	99	97	91	94	81
S_{10}	100	100	98	92	95	81
S_{11}	94	76	79	44	73	38
S_{12}	94	75	74	24	70	37
S_{13}	98	95	95	85	93	77
S_{14}	81	49	64	26	72	39
S_{15}	98	96	96	89	94	78
S_{16}	74	24	72	24	71	21

Table 2(b). Prediction accuracy of the multivariate Markov model (Part 2).

Noise	Prediction accuracy in %			
	10%		20%	
	r%	R%	r%	R%
Initial state				
S_1	62	17	63	14
S_2	90	65	80	37
S_3	66	21	65	20
S_4	87	63	77	42
S_5	68	20	63	20
S_6	71	35	62	14
S_7	89	63	80	40
S_8	90	68	81	41
S_9	90	66	79	43
S_{10}	90	65	81	36
S_{11}	67	17	66	15
S_{12}	71	21	64	14
S_{13}	88	64	79	37
S_{14}	66	23	62	15
S_{15}	89	63	79	36
S_{16}	66	23	61	13

$T = 100$ based on the truth table. The sequences are then used to build

- (i) a PBN by the COD method and
- (ii) a multivariate Markov model by our proposed method.

For each of the noise levels, the average values of r and R for the 16 sequences with different initial states and the average computational time for both PBN and multivariate Markov model are reported in Table 3.

In the experiments, although the prediction accuracy of our model in all cases is lower than that of PBNs, the efficiency of our model is much better for such small number of genes. It is expected that the

 Table 3. Comparison on average prediction accuracy with $T = 100$.

Noise %	Prediction accuracy in %					
	PBN			Multivariate model		
	r%	R%	Time (sec)	r%	R%	Time (sec)
0.0	100	100	6.63	91	74	0.25
2.5	97	89	6.52	85	62	0.26
5.0	93	80	6.55	83	58	0.27
10.0	86	60	6.60	78	43	0.24
20.0	76	35	7.32	71	28	0.29

computational times of using our model are significantly less than those using PBNs when the number of genes is large. In the PBNs computational process, we encounter some empty sets of \mathcal{F}_i because all the corresponding $c_j^{(i)}$ are equal to zero. We have 3, 4, 4, 5 and 4 unsolvable cases out of 16 trios in noise free, 2.5% noise, 5% noise, 10% noise and 20% noise cases respectively. However, we do not have this kind of problem in our model. Furthermore, the results on our model represent fair high prediction accuracy in all cases. We remark that the expected accuracy rate for r and R are given as follows:

$$\begin{aligned}
 E[r] &= E \left[\frac{1}{nT} \times \sum_{i=1}^n \sum_{t=1}^T \delta_t^{(i)} \times 100\% \right] \\
 &= \frac{1}{nT} \times \sum_{i=1}^n \sum_{t=1}^T E(\delta_t^{(i)}) \times 100\% \\
 &= \frac{1}{2} \times 100\% = 50\%
 \end{aligned}$$

and

$$\begin{aligned}
 E[R] &= E \left[\frac{1}{T} \times \sum_{t=1}^T \delta_t \times 100\% \right] \\
 &= \frac{1}{T} \times \sum_{t=1}^T \left(1 \cdot \frac{1}{2^n} + 0 \right) \times 100\% \\
 &= \frac{1}{T} \times \sum_{t=1}^T \left(1 \cdot \frac{1}{16} + 0 \right) \times 100\% \\
 &= \frac{1}{16} \times 100\% = 6.25\%
 \end{aligned}$$

where n is the number of sequences. We also tried the case with $T = 1000$, and the average prediction results are shown in Table 4.

 Table 4. Comparison on average prediction accuracy with $T = 1000$.

Noise %	Prediction accuracy in %					
	PBN			Multivariate model		
	r%	R%	Time (sec)	r%	R%	Time (sec)
0.0	100	100	58.69	91	75	0.31
2.5	96	89	58.39	87	65	0.33
5.0	93	78	58.62	82	51	0.33
10.0	83	57	59.47	80	46	0.31
20.0	69	29	60.02	70	28	0.31

Table 5. Prediction accuracy of the multivariate Markov model.

n	$r\%$	$\frac{r\%}{E[r]\%}$	Time (sec)
<i>Noise = 0</i>			
8	73.29	1.47	0.49
16	59.27	1.19	0.95
32	60.33	1.21	2.17
64	61.58	1.23	4.80
128	62.73	1.25	12.39
<i>Noise = 2.5%</i>			
8	74.21	1.48	0.48
16	59.21	1.18	0.95
32	60.36	1.21	2.17
64	61.56	1.23	4.79
128	62.68	1.25	12.29
<i>Noise = 5%</i>			
8	71.06	1.42	0.48
16	59.14	1.18	0.96
32	60.44	1.21	2.05
64	61.70	1.23	4.81
128	62.76	1.26	12.75
<i>Noise = 10%</i>			
8	69.01	1.38	0.48
16	59.14	1.18	0.96
32	60.40	1.21	2.05
64	61.62	1.23	5.02
128	62.72	1.25	12.47
<i>Noise = 20%</i>			
8	63.60	1.27	0.46
16	58.94	1.18	0.95
32	60.43	1.21	2.05
64	61.63	1.23	5.06
128	62.81	1.26	12.34

We notice that the prediction accuracy of PBNs is decreased by 2.2% in both r and R , while the prediction results of our model are increased by 0.4% and 0% in r and R respectively. The computational time is slightly increased to about 0.32 seconds for our model. While for a PBN, the computational time is increased to about 59.04 seconds.

6.2. Test with synthetic data II

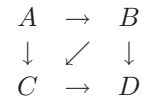
In this subsection, we compare the average prediction accuracy rates r and R as well as the average computational time for various number of sequences, $n = 8, 16, 32, 64$ and 128 . In PBNs, even in the case $n = 8$, the number of distinct predictors to be computed is $2^{2^8} \approx 1.1579e + 077$. If we are to compute all of them in a PC, it requires approximately

3.36×10^{65} years. Therefore, we only report results on our model where the sequence generating method is similar to that in the previous subsection. Here we choose the length $T = 100$ and repeat each case 50 times. The results are then reported in Table 5. According to Table 5, the prediction accuracy only decreases slightly even when the size of the network increases linearly.

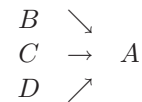
In this model, we notice that our model parameter estimation methods are very efficient. Actually, the computational time increases linearly with respect to the number of sequences in the network. When $n \geq 16$, the prediction accuracy increases as the size of n increase. This improvement may be caused by the increased sized of training data. Moreover, the prediction accuracy is quite close with various noise level. The main reason is that the length of sequence $T (= 100)$ is limited relative to the number of possible networks ($n2^{2^n}$) and it is impossible to obtain the true networks from the training data.

6.3. Test with synthetic data III

In the following, we give an example of a PBN having four vertices. Firstly, we assume the PBN has the following structure:



and



The relationship between each sequence can be represented by following transition matrices:

$$\overbrace{\begin{pmatrix} 0.37 & 0.48 \\ 0.63 & 0.52 \end{pmatrix}}^{A \rightarrow B},$$

$$\overbrace{\begin{pmatrix} 0.40 & 0.49 & 0.24 & 0.38 & 0.34 & 0.03 & 0.06 & 0.68 \\ 0.60 & 0.51 & 0.76 & 0.62 & 0.66 & 0.97 & 0.94 & 0.32 \end{pmatrix}}^{B, C, D \rightarrow A},$$

$$\overbrace{\begin{pmatrix} 0.50 & 0.47 & 0.46 & 0.76 \\ 0.50 & 0.53 & 0.54 & 0.24 \end{pmatrix}}^{A, B \rightarrow C}$$

Table 6. The Comparison on average prediction accuracy.

Noise %	$r\%$	$R\%$	Time (sec)
<i>PBN</i>			
0.0	60.91	14.62	59.31
2.5	59.71	13.20	59.24
5.0	59.55	12.58	59.42
10.0	58.73	12.26	59.41
20.0	57.06	10.77	59.48
<i>Multivariate Markov model</i>			
0.0	58.76	12.98	0.33
2.5	58.03	12.26	0.33
5.0	57.80	11.77	0.33
10.0	57.15	11.52	0.31
20.0	55.39	9.83	0.32

and

$$\overbrace{\begin{pmatrix} 0.44 & 0.41 & 0.54 & 0.44 \\ 0.56 & 0.59 & 0.46 & 0.56 \end{pmatrix}}^{B,C \rightarrow D}$$

Based on the above matrices, we generate 4 sequences with $T = 1000$. We repeat this process 10 times and the average results are shown in Table 6.

For instance, in the absence of noise, the average value of λ_{ij} is

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1n} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n1} & \lambda_{n2} & \dots & \lambda_{nn} \end{pmatrix} = \begin{pmatrix} 0.2 & 0.7 & 0.1 & 0.0 \\ 0.4 & 0.5 & 0.1 & 0.0 \\ 0.4 & 0.2 & 0.3 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{pmatrix}.$$

In the presence of cycle, our average results in all cases are not far from the PBNs results but computational time of PBNs is about 180 times of our model. Furthermore, the resulting Λ is not a diagonal dominant matrix, which showed a strong cycle relationship in this four sequences.

6.4. Test with the gene expression data of yeast

Genome transcriptional analysis is important in medicine, etiology and bioinformatics. One of the

applications of genome transcriptional analysis is used for eukaryotic cell cycle in yeast. The fundamental periodicity in eukaryotic cell cycle includes the events of DNA replication, chromosome segregation and mitosis. Hartwell and Kastan¹⁴ suggested that improper cell cycle regulation leads to genomic instability, especially in etiology of both hereditary and spontaneous cancers, instances in Wang *et al.*,³¹ Hall and Peters.¹³ Eventually, it is believed to play one of the important roles in the etiology of both hereditary and spontaneous cancers. Genome transcriptional analysis helps in exploring the cell cycle regulation and the mechanism behind the cell cycle. Raymond *et al.*²⁴ examined the presence of cell cycle-dependent periodicity in 6220 transcripts and found that cell cycle appears in about 7% of transcripts. Those transcripts are then extracted for further examination. When the time course was divided into early G1, late G1, S, G2 and M phase, the result showed that more than 24% of transcripts are directly adjacent to other transcripts in the same cell cycle phase. Further investigating result on those transcripts also indicated that more than half are affected by more than one cell cycle-dependent regulatory sequences.

The data set used in our study is the selected set from Yeung and Ruzzo.³⁴ In the discretization, if an expression level is above (below) its standard deviation from the average expression of the gene, it is over-expressed (under-expressed) and the corresponding state is 1 (0), we intend to find out this relationship from 213 well-known yeast transcripts with cell cycle in order to illustrate the ability of our proposed model. This problem can be solved by a PBN theoretically. However, there are two problems in using PBNs in practice. First, it is clear that the method of COD is commonly used to estimate the probabilities of each predictor $c_g^{(d)}$ for transcript d . Unfortunately, owing to limited time points of the expression level of each gene (there are only 17 time points for the yeast data set), it is almost impossible to find a value of $c_g^{(d)}$ which is strictly greater than that of the best estimation in the absence of any conditional variables. Therefore, most of the transcripts do not have any predictor, which makes it impossible to estimate the PBN parameters. Moreover, PBN seems to be unable to model a set of genes when n is quite large. Friedman *et al.*¹² suggested Bayesian networks could infer a genetic network successfully, but it is unable to infer a genetic network with cell cycle relationship. Ott *et al.*²³ also suggested

that even if in an acyclic genetic network with constraints situation, the number of genes in Bayesian networks should not be greater than 40 if BNRC (Bayesian Network and Nonparametric Regression Criterion) score are used. Later, Kim *et al.*²¹ proposed a dynamic Bayesian network which can construct of cyclic regulations for medium time-series, but it seems to be not practical to handle a large networks. Here, we use the multivariate Markov model for training the yeast data set. The construction of the multivariate Markov chain models for such data set only requires around 0.1 second. This showed that our proposed method is very efficient. In our study, we assume that there is no any prior knowledge about the genes. Therefore, in the construction of the multivariate Markov chain models, each target gene can be

Table 7. Result of our multivariate Markov model.

No.	Name of target transcript	Cell cycle phase	Length of cell cycle	Related transcripts (its phase λ_{ij} , level of influence)
(1)	YDL101c	late G1	1	YMR031c (1,1.00,1.00)
(2)	YKL127W	S	1	YML027w (2,1.00,0.75)
(3)	YKL113c	late G1	2	YDL018c (2,0.50,0.50) YOR315w (5,0.50,0.50)
(4)	YPL127c	late G1	2	YDL101c (2,0.33,0.38) YML027w (2,0.33,0.39) YJL079c (5,0.33,0.38)
(5)	YLR121c	late G1	3	YPL158c (1,0.33,0.42) YDL101c (2,0.33,0.43) YKL069w (4,0.33,0.43)
(6)	YKL069W	G2	3	YLR274w (1,0.50,0.50) YER001w (3,0.50,0.50)
(7)	YLR015w	early G1	4	YKL113c (2,1.00,0.88)
(8)	YGR279c	M	4	YPL158c (1,1.00,0.80)

Table 8. Prediction results.

Length of cell cycle phases required	Number of occurrence in this type of cell cycle	Average prediction accuracy	Example in Table 7
1	5%	86%	(1),(2)
2	9%	87%	(3),(4)
3	9%	83%	(5),(6)
4	70%	86%	(7),(8)

related to other genes. Based on the values of λ_{ij} in our model, we can determine the occurrence of cell cycle in j th transcript, i.e., presence of an inter-relationship of any j th transcript in a set of transcripts. Using the multivariate Markov chain models, we find that 93% of transcripts are possibly involved in some cell cycles. Some of the results are shown in Table 7.

In Table 7, the first column indicates the ID number of data set we display. The second column shows the name of target transcript. The third column shows which phase the target gene belongs to. The fourth column shows the most possible cell cycle length of the target transcript. The last column displays the name of required transcripts for predicting the target transcript, the corresponding phase of required transcripts, their corresponding weightings λ_{ij} in the model, as well as an estimated value of the level of influence from related transcript to the target transcript. Although the level of influence can be estimated based on our model parameters, its computational cost in the PBN method increases exponentially with respect to the value of n . We find in Table 7 that the weighting λ_{ij} provides a reasonable measure for the level of influence. Therefore the proposed method can estimate the level of influence very efficiently. We present the prediction results of different lengths of cell cycle for the whole data set in the following table and the results show that the performance of the model is effective.

The average value of r for the 271 gene expression sequences is given by 85%.

7. Concluding Remarks

In this paper, we proposed a multivariate Markov model for a PBN. Efficient estimation methods are presented to obtain the model parameters. Methods for recovering the structure and rules of a PBN are also

illustrated in detail. Numerical experiments on both synthetic data and gene expression data of yeast are given to demonstrate the effectiveness of our proposed model. Our proposed model gives satisfactory results especially in the presence of cell cycle data. The model can be easily extended to the case where gene expression data have more than two levels. In this case, we still have Theorems 1 and 2. Moreover the estimation and prediction methods can still be applied. Another direction for further research is to consider a higher-order multivariate model, and to develop estimation and prediction methods for the model.

In bioinformatics, it is often the case that the bioinformaticians possess some background knowledge on the gene expression data set that can be useful in modeling genetic networks. Background knowledge in the form of functional pathways over some of genes can be incorporated as supervision to the construction of multivariate Markov chain model. Let $S^{(jk)}$ denote the prior transition probability matrix for the estimation of $P^{(jk)}$ in Sec. 5. Then, the estimate $P_e^{(jk)}$ of $P^{(jk)}$ is given by:

$$P_e^{(jk)} = w_{jk}S^{(jk)} + (1-w_{jk})\hat{P}^{(jk)}, \quad j, k = 1, 2, \dots, n,$$

where $0 \leq w_{jk} \leq 1$, for each $j, k = 1, 2, \dots, n$. For the estimation of the \hat{P}_e -matrix, we also need to estimate the parameters λ_{jk} . From Theorem 1, we have that

$$\begin{pmatrix} \lambda_{11}P_e^{(11)} & \lambda_{12}P_e^{(12)} & \dots & \lambda_{1n}P_e^{(1n)} \\ \lambda_{21}P_e^{(21)} & \lambda_{22}P_e^{(22)} & \dots & \lambda_{2n}P_e^{(2n)} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{n1}P_e^{(n1)} & \lambda_{n2}P_e^{(n2)} & \dots & \lambda_{nn}P_e^{(nn)} \end{pmatrix} \hat{\mathbf{V}} \approx \hat{\mathbf{V}},$$

where the approximation sign “ \approx ” means that the probability vector on the left hand side is approximated by the vector in the right hand side in the component-wise sense. Let $\tilde{\lambda}_{jk}^1 = \lambda_{jk}w_{jk}$ and $\tilde{\lambda}_{jk}^2 = \lambda_{jk}(1-w_{jk})$. Then, it is easy to check that $\tilde{\lambda}_{jk}^1 + \tilde{\lambda}_{jk}^2 = \lambda_{jk}$, for each $j, k = 1, 2, \dots, n$. We notice that the estimation of λ_{jk} and w_{jk} is equivalent to the estimation of $\tilde{\lambda}_{jk}^1$ and $\tilde{\lambda}_{jk}^2$. Then, the above can be written in the following form:

$$\begin{pmatrix} \tilde{\lambda}_{11}^1 S^{(11)} + \tilde{\lambda}_{11}^2 \hat{P}^{(11)} & \dots & \tilde{\lambda}_{1n}^1 S^{(1n)} + \tilde{\lambda}_{1n}^2 \hat{P}^{(1n)} \\ \tilde{\lambda}_{21}^1 S^{(21)} + \tilde{\lambda}_{21}^2 \hat{P}^{(21)} & \dots & \tilde{\lambda}_{2n}^1 S^{(2n)} + \tilde{\lambda}_{2n}^2 \hat{P}^{(2n)} \\ \vdots & \vdots & \vdots \\ \tilde{\lambda}_{n1}^1 S^{(n1)} + \tilde{\lambda}_{n1}^2 \hat{P}^{(n1)} & \dots & \tilde{\lambda}_{nn}^1 S^{(nn)} + \tilde{\lambda}_{nn}^2 \hat{P}^{(nn)} \end{pmatrix} \hat{\mathbf{V}} \approx \hat{\mathbf{V}}$$

The estimation problem can be formulated similar to that in Sec. 5. Our future research work is to demonstrate the usefulness of the semi-supervised approach in modeling genetic networks.

Acknowledgments

The authors would like to thank two anonymous referees and Prof. Hojjat Adeli for their helpful comments and suggestions in revising the paper.

Research supported in part by RGC Grant Nos. HKU 7126/02P, 7130/02P, 7046/03P, 7035/04P, 7035/05P and HKU CRCG Grant Nos. 10203919, 10203907, 10204437, 10203501.

References

1. T. Akutsu, S. Miyano and S. Kuhara, Inferring qualitative relations in genetic networks and metabolic arrays, *Bioinformatics* **16** (2000) 727–734.
2. J. Bower, *Computational Modeling of Genetic and Biochemical Networks* (MIT Press, Cambridge, M.A., 2001).
3. A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences* (Academic Press, New York, 1979).
4. J. Bodnar, Programming the Drosophila embryo, *J. Theoret. Biol.* **188** (1997) 391–445.
5. Y. Chen, E. Dougherty and M. Blittner, Ratio-based decisions and the quantitative analysis of cDNA microarray images, *J. Biomed. Opt.* **2** (1997) 364–374.
6. W. Ching, E. Fung and M. Ng, A multivariate Markov chain model for categorical data sequences and its applications in demand predictions, *IMA Journal of Management Mathematics* **13** (2002) 187–199.
7. W. Ching, E. Fung and M. Ng, Building genetic networks in gene expression patterns, *Lecture Notes in Computer Science* (Springer) **3177** (2004) 17–24.
8. V. Chvatal, *Linear Programming* (Freeman, New York, 1983).
9. H. de Jong, Modeling and simulation of genetic regulatory systems: A literature review, *J. Comput. Biol.* **9** (2002) 69–103.
10. E. Dougherty, S. Kim and Y. Chen, Coefficient of determination in nonlinear signal processing, *Signal Processing*, **80** (2000) 2219–2235.
11. S. Fang and S. Puthenpura, *Linear Optimization and Extensions* (Prentice-Hall, Englewood Cliffs, NJ, 1993).
12. N. Friedman, M. Linial, I. Nachman and D. Pe’er, Using Bayesian networks to analyze expression data, *Journal of Computational Biology* **7**(3–4) (2000) 601–620.

13. M. Hall and G. Peters, Genetic alterations of cyclins, cyclin-dependent kinases, and Cdk inhibitors in human cancer, *Adv. Cancer Res.* **68** (1996) 67–108.
14. L. H. Hartwell and M. B. Kastan, Cell cycle control and cancer, *Science* **266** (1994) 1821–1828.
15. S. Huang and D. E. Ingber, Shape-dependent control of cell growth, differentiation, and apoptosis: Switching between attractors in cell regulatory networks, *Exp. Cell Res.* **261** (2000) 91–103.
16. S. Kauffman, Metabolic stability and epigenesis in randomly constructed gene nets, *J. Theoret. Biol.* **22** (1969) 437–467.
17. S. Kauffman, Homeostasis and differentiation in random genetic control networks, *Nature* **224** (1969) 177–178.
18. S. Kauffman, *The origin of orders* (Oxford University Press, New York, 1993).
19. J. Khan, R. Simon, M. Bittner, Y. Chen, S. B. Leighton, T. Pohida, P. D. Smith, Y. Jiang, G. C. Gooden, J. M. Trent and P. S Meltzer, Gene expression profiling of alveolar rhabdomyosarcoma with cDNA microarrays, *Cancer Res.* **58** (1998) 5009–5013.
20. S. Kim, E. Dougherty, Y. Chen, K. Sivakumar, P. Meltzer, J. Trent and M. Bittner, Multivariate measurement of gene expression relationships, *Genomics* **67** (2000) 201–209.
21. S. Kim, S. Imoto and S. Miyano, Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data, in *Proc. 1st Computational Methods in Systems Biology, Lecture Note in Computer Science* **2602** (Springer-Verlag, 2003), pp. 104–113.
22. L. Mendoza, D. Thieffry and E. R. Alvarez-Buylla, Genetic control of flower morphogenesis in *Arabidopsis thaliana*: A logical analysis, *Bioinformatic* **15** (1999) 593–606.
23. S. Ott, S. Imoto and S. Miyano, Finding optimal models for small gene networks, *Pacific Symposium on Biocomputing* **9** (2004) 557–567.
24. J. Raymond, J. Michael, A. Elizabeth, S. Lars et al., A genome-Wide transcriptional analysis of the mitotic cell cycle, *Molecular Cell* **2** (1998) 65–73.
25. S. Ross, *Introduction to Probability Models* (Academic Press, San Diego, CA, 2000).
26. I. Shmulevich, E. Dougherty, S. Kim and W. Zhang, Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks, *Bioinformatics* **18** (2002a) 261–274.
27. I. Shmulevich, E. Dougherty, S. Kim and W. Zhang, Control of stationary behavior in Probabilistic Boolean networks by means of structural intervention *J. Biological Systems* **10** (2002b) 431–445.
28. I. Shmulevich, E. Dougherty, S. Kim and W. Zhang, From Boolean to probabilistic Boolean networks as models of genetic regulatory networks, in *Proc. IEEE* **90** (2002c), pp. 1778–1792.
29. P. Smolen, D. Baxter and J. Byrne, Mathematical modeling of gene network, *Neuron* **26** (2000) 567–580.
30. A. Tsodikov, A. Szabo and D. Jones, Adjustments and measures of differential expression for microarray data, *Bioinformatics* **18**(2) (2002) 251–260.
31. T. C. Wang, R. D. Cardiff, L. Zukerberg, E. Lees, A. Arnold and E. V. Schmidt, Mammary hyperplasia and carcinoma in MMTV-cyclin D1 transgenic mice, *Nature* **369** (1994) 669–671.
32. L. F. A. Wessels, E. P. Van Someren and M. J. T. Reinders, A comparison of genetic network models, in *Pacific Symposium on Biocomputing* **6** (2001) pp. 508–519 .
33. A. Wuensche, Genomic regulation modeled as a network basins of attraction, in *Pac. Symp. Biocomput.* **3** (1998), pp. 89–102.
34. K. Yeung and W. Ruzzo, An empirical study on principal component analysis for clustering gene expression data, *Bioinformatics* **17** (2001) 763–774.