Sun Yong Kim
is a graduate student at the laboratory of DNA analysis, Human Genome Centre, Institute of Medical Science, University of Tokyo, Japan. Her current interests include modelling gene networks from time series microarray gene expression data using statistical methods, including DBNs.

Seiya Imoto
is currently a research associate at the laboratory of DNA analysis, Human Genome Centre, Institute of Medical Science, University of Tokyo. His current research interests cover analysis of high-dimensional data with complex structure using nonlinear statistical methods, as well as development model selection criteria from an information theoretic or a Bayesian statistics approach, and analysis of cDNA microarray gene expression data.

Satoru Miyano
is a professor at the Human Genome Centre, Institute of Medical Science, University of Tokyo. His current interests include computational gene network inference methods, modelling and simulation of biological systems, and computational knowledge discovery. He is on the Editorial Board of *Bioinformatics*, *J. Bioinformatics and Computational Biology* and *Theoretical Computer Science*, and is the Chief Editor of *Genome Informatics*.

***Keywords:** microarray, gene networks, DBNs*

SunYong Kim,
Human Genome Center,
Institute of Medical Science,
University of Tokyo,
4-6-1 Shirokanedai, Minato-ku,
Tokyo 108-8639, Japan

Tel: +81 3 5449 5615
Fax: +81 3 5449 5442
E-mail: sunk@ims.u-tokyo.ac.jp

Inferring gene networks from time series microarray data using dynamic Bayesian networks

Sun Yong Kim, Seiya Imoto and Satoru Miyano

Date received (in revised form): 23rd June 2003

Abstract

Dynamic Bayesian networks (DBNs) are considered as a promising model for inferring gene networks from time series microarray data. DBNs have overtaken Bayesian networks (BNs) as DBNs can construct cyclic regulations using time delay information. In this paper, a general framework for DBN modelling is outlined. Both discrete and continuous DBN models are constructed systematically and criteria for learning network structures are introduced from a Bayesian statistical viewpoint. This paper reviews the applications of DBNs over the past years. Real data applications for *Saccharomyces cerevisiae* time series gene expression data are also shown.

INTRODUCTION

The development of microarray technology produces a huge amount of gene expression data and provides an innovative perspective for whole genome analyses. The estimation of a gene network from cDNA microarray gene expression data is one of the most important computational topics. Several methods have been proposed for modelling gene networks including: Boolean networks,[1,2] Bayesian networks (BNs)[3–5] and differential equations.[6,7] In particular, researchers have paid great attention to BNs, which model causal relationships between variables based on probabilistic measure. Since microarray data are usually very noisy, the use of statistical methods is expected to be effective for extracting useful information from such noisy data. Friedman *et al.*[3] proposed both a discrete BN model and a continuous BN model based on a linear regression for modelling gene networks. Imoto *et al.*[4,5] succeeded in employing a non-parametric regression for capturing even non-linear relationships between genes.

Although the above methods are effective to some degree, BNs have a limitation that no cycles are allowed. This can be a serious problem since real gene networks have cyclic regulatory pathways including feedback loops. When we have time series microarray data, the use of dynamic Bayesian networks (DBNs) is a promising alternative, since DBNs can treat time delay information and can construct cyclic networks. DBNs have been used in the field of signal processing and were recently introduced into the analysis of time series microarray data. Friedman *et al.*[8] first applied DBNs to the analysis of gene networks. They constructed a discrete DBN model and used the BDe[9] metric for learning networks. Smith *et al.*[10,11] and Ong *et al.*[12] also used discrete models. An interesting point of Ong *et al.* is that they imported biological knowledge into the modelling of network structures. Their target organism, *Escherichia coli*, is already known to have sets of genes, called operons, which are transcribed together into mRNA. Reflecting this information, they added some nodes representing operons to the network and restricted edge directions. Although discrete models have

Joint probability

Gene network

Time series microarray data

Dynamic Bayesian network

Conditional probability

Time dependency

some advantages such as robustness, simplicity of learning and non-linearity, discretisation often tends to be a problem for the following two reasons. First, discretisation might cause information loss. Secondly, the threshold value for discretisation must be chosen very carefully since resulting networks will be affected by this value. To avoid discretisation, Kim *et al.*[13] defined a continuous DBN and non-parametric regression model to capture more than linear dependencies.

This paper reviews the methodology of estimating gene networks from time series microarray data using DBN models. A general theory of DBN models is introduced first, and discrete and continuous models are then elicidated. Information criteria for learning unknown network structures from a Bayesian statistical viewpoint are derived. The methods in Friedman *et al.*,[8] Smith *et al.*,[10,11] Ong *et al.*[12] and Kim *et al.*[13] will be presented in this framework. The effectiveness of DBN models through the analysis of *S. cerevisiae* microarray data will be shown.[14]

DBN MODEL

DBNs can be viewed as an extension of BNs. In contrast to BNs that are based on static data, DBNs use time series data for constructing causal relationships among random variables. In this section, we describe a DBN model under a general framework.

Suppose that we have n microarrays and each microarray measures expression levels of p genes. The microarray data, then, can be summarised as an $n \times p$ matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$ whose ith row vector $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ corresponds to a gene expression level vector measured at time t. Note that x_{ij} is considered as an observation from a random variable X_{ij}. In DBN modelling, the process of model construction can be divided into two steps. First, the DBN models assume a time dependency. Note that, in general, edges in a time slice can be allowed, but in this paper, models are assumed within which the state vector of time i depends only on that of time $i-1$. Figure 1 shows this relationship as a directed acyclic graph. Therefore, the joint probability can be decomposed as:

$$P(X_{11}, \ldots, X_{np}) = P(\boldsymbol{X}_1)P(\boldsymbol{X}_2|\boldsymbol{X}_1) \times \ldots \times P(\boldsymbol{X}_n|\boldsymbol{X}_{n-1}) \quad (1)$$

where $\boldsymbol{X}_i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ is a p-dimensional random variable vector.

Next, we consider the gene regulations described in the right side of Figure 1. The gene regulations can be modelled through the construction of $P(\boldsymbol{X}_i|\boldsymbol{X}_{i-1})$ for $i = 2, \ldots, n$. We assume that gene j has q_j genes as did its parents. As is shown in Figure 1, the network structure is assumed to be stable through all time points. Furthermore, according to the time dependency, only forward edges, ie edges from time $i-1$ to i, are allowed in these networks. Hence DBNs can model cycles, as is shown in Figure 2. Under these conditions, the conditional probability $P(\boldsymbol{X}_i\,|\,\boldsymbol{X}_{i-1})$ can also be decomposed into the product of conditional probabilities of each gene given its parent genes:

$$P(\boldsymbol{X}_i|\boldsymbol{X}_{i-1}) = P(X_{i1}|\boldsymbol{P}_{i-1,1}) \times \ldots \times P(X_{ip}|\boldsymbol{P}_{i-1,p}) \quad (2)$$

where $\boldsymbol{P}_{i-1,j} = (P^{(j)}_{i-1,1}, \ldots, P^{(j)}_{i-1,q_j})^{\mathrm{T}}$ is a random variable vector of parent genes of jth gene at time $i-1$.

Equations (1) and (2) hold when we use density or probability functions instead of probabilistic measure. We then obtain a DBN model in the form:

$$f(x_{11}, \ldots, x_{np}) = \prod_{i=1}^{n}\prod_{j=1}^{p} g_j(x_{ij}|\boldsymbol{p}_{i-1,j})$$

where $\boldsymbol{p}_{oj} = \phi$.

In statistics, we parameterise f by a parameter vector θ and transfer the construction of f into the estimation of θ.

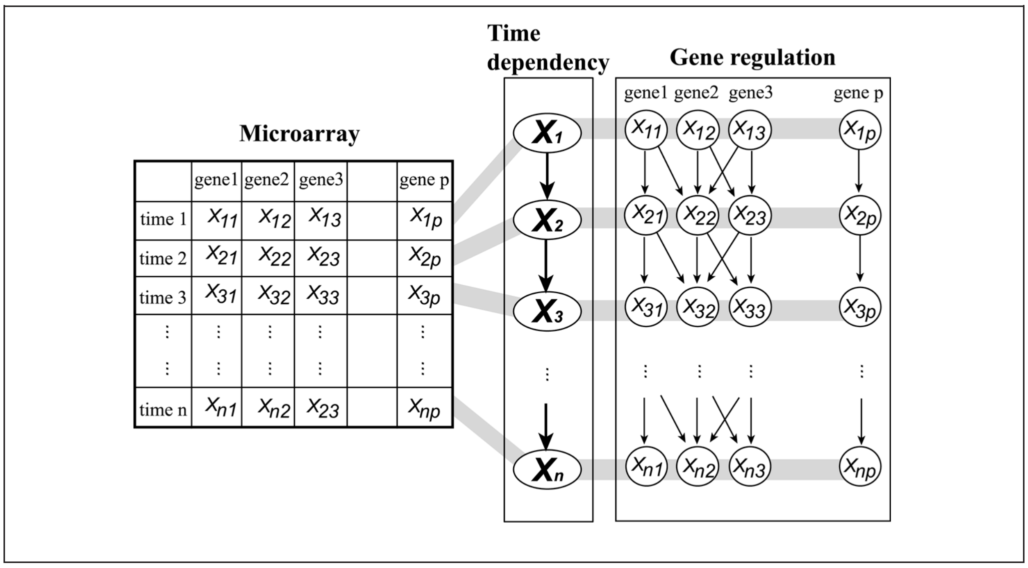

Figure 1: Graphical view of a dynamic Bayesian network model

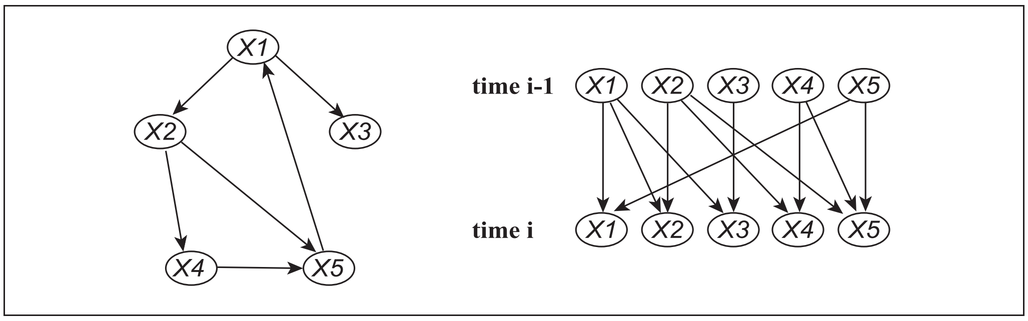

Figure 2: Example of a network containing a cyclic regulation. The network (left) contains a cycle $X_1 \rightarrow X_2 \rightarrow X_4 \rightarrow X_5 \rightarrow X_1$. A Bayesian network model cannot treat such a network. On the other hand, the dynamic Bayesian network can construct a cyclic regulation by dividing states of a gene by time points (right)

Discrete model

Although microarray data are measured as continuous data, discretisation is sometimes applied in order to remove noise. Then a discrete DBN is used for estimating gene networks. Let $U = \{u_1, \ldots u_m\}$ be a finite set of discrete values and $I_1, \ldots, I_m$ be regions satisfying $\Upsilon_{l=1}^{m} I_l = \mathscr{R}$ and $I_i II_j = \phi$ $(i \neq j)$. Here $\mathscr{R}$ is the set of real values. An expression value x_{ij} is then transformed to u_l when $x_{ij} \in i_l$. The values $g_j(x_{ij}|\boldsymbol{p}_{i-1,j})$ themselves are considered as parameters, that is $\theta_{jkl} = P(X_{ij} = u_l|\boldsymbol{P}_{i-1,j} = \boldsymbol{u}_{jk})$, where $\boldsymbol{u}_{jk}$ is the kth entry of the state table of parents of the jth gene. For example, suppose that we discretise the expression values into two classes and that the jth gene has two parents. That is, the expression value x_{ij} is transformed into 0 or 1 and the state table will have four entries, $\boldsymbol{u}_{j1} = (0, 0)$, $\boldsymbol{u}_{j2} = (0, 1)$, $\boldsymbol{u}_{j3} = (1, 0)$, $\boldsymbol{u}_{j4} = (1, 1)$. Then $f(x_{11}, \ldots, x_{np}; \boldsymbol{\theta})$ can be modelled as a multinomial distribution function:

Multinomial distribution

$$f(x_{11}, \ldots, x_{np}; \boldsymbol{\theta}) = \prod_{j=1}^{p} \prod_{k=1}^{Q_j} \prod_{l=1}^{m} \theta_{jkl}^{N_{jkl}} \quad (3)$$

where $\boldsymbol{\theta} = (\theta_{111}, \ldots, \theta_{pQpm})^{\mathrm{T}}$, N_{jkl} indicates the number of observations satisfying $x_{ij} = u_l$ and $\boldsymbol{p}_{i-1,j} = \boldsymbol{u}_{jk}$ for $i = 2, \ldots n$, and $Q_j = m^{qj}$ is the number of entries of the state table of parents of the jth gene.

Normal density

Recall that microarray data need to be discretised while using discrete models. In general, discretisation is performed as follows: Let $t_0, \ldots, t_m$ be thresholds for discretisation satisfying $t_0 = \min_{i,j} x_{ij} < t_1 < \ldots < t_m = \max_{i,j} x_{ij}$ and x_{ij} will be classified to u_l if $t_{l-1} < x_{ij} < t_l$. Note that several methods, such as the *k*-means algorithm, have been investigated for discretising microarray data (see, for example, Friedman and Goldszmidt[15] and Pe'er *et al.*)[16] Friedman *et al.*[8] do not provide the details of discretisation. They discretised *Sacccharomyces cerevisiae* gene expression data into three classes, however: over-expressed, under-expressed and normal, depending on whether the expression rate was significantly greater than, lower than and similar to control, respectively. Smith *et al.*[10,11] analysed artificial data generated from a simulator of a communication system of birds. They discretised their data into four classes and set the thresholds as $t_1 = r(x_{ij})/4$, $t_2 = r(x_{ij})/2$ and $t_3 = 3r(x_{ij})/4$, where $r(x_{ij}) = \max_{i,j} x_{ij} - \min_{i,j} x_{ij}$. This discretisation seems to be suitable for their data. Further discussion is needed when we apply this discretisation method to real microarray data in practice, however. Ong *et al.*[12] used *E. coli* microarray data discretised into two classes. An expression value x_{ij} is transformed to u_1 if $x_{ij} > x_{i-1,j}$ or u_2 otherwise. This discretisation could be sensitive to noise when the expression level changes in a narrow range.

Discretisation

Continuous model

When we treat microarray data as continuous values, $g_j(x_{ij}|\boldsymbol{p}_{i-1,j}; \theta_j)$ can be modelled as a normal density function:

$$g_j(x_{ij}|\mathbf{p}_{i-1,j}; \boldsymbol{\theta}_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left\{-\frac{[x_{ij} - m(\boldsymbol{p}_{i-1,j})]^2}{2\sigma_j}\right\}$$

where θ_j is a parameter vector in $g_j(\cdot)$ and $m(\boldsymbol{p}_{i-1,j})$ is a regression function from $\mathscr{R}^{q_j}$ to $\mathscr{R}$. For example, if we define

$$m(\boldsymbol{p}_{i-1,j}) = \beta_1^{(j)} p_{i-1,1}^{(j)} + \ldots + \beta_{q_j}^{(j)} p_{i-1,q_j}^{(j)}$$

we obtain a linear DBN model, where $\beta_1, \ldots, \beta_{q_j}$ are parameters.

Non-parametric regression

B-splines

There is no guarantee that the linear models can approximate the relationships between genes, however. For capturing even non-linear relationships between genes, Kim *et al.*[13] used a non-parametric regression model based on *B*-splines:

$$m(\boldsymbol{p}_{i-1,j}) = \sum_{m=1}^{M_{j1}} \gamma_{m1}^{(j)} b_{m1}^{(j)}(p_{i-1,1}^{(j)}) + \ldots + \sum_{m=1}^{M_{jq_j}} \gamma_{mq_j}^{(j)} b_{mq_j}^{(j)}(p_{i-1,q_j}^{(j)})$$

where $\gamma_{1k}^{(j)}, \ldots, \gamma_{M_{jk}}^{(j)} k$ are coefficient parameters and $\{b_{1k}^{(j)}(\cdot), \ldots, b_{M_{jk}k}^{(j)}(\cdot)\}$ is a prescribed set of *B*-splines.

CRITERION FOR LEARNING NETWORKS

By using DBN models, we can model a gene network from time series microarray data, when we know the true relationships among genes completely. Many parts of the true gene network are

Posterior probability of the network

still unknown and need to be estimated from microarray data, however. Hence, construction of a criterion for evaluating the goodness of the specified model is an essential point of gene network modelling. Under a Bayesian statistics framework, we can choose an optimal network by maximising the posterior probability of the network. The posterior probability of the network G is given by:

Marginal likelihood

$$P(G|\boldsymbol{X}) = P(G, \boldsymbol{X})/P(\boldsymbol{X})$$

where

$$P(G, \boldsymbol{X}) = \int P(G, \mathbf{X}, \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$$

$$= P(G) \int P(\mathbf{X}|\boldsymbol{\theta}, G) P(\boldsymbol{\theta}|G) d\boldsymbol{\theta}$$

$$P(\mathbf{X}) = \sum_{G \in \Omega} P(G, \mathbf{X})$$

Here Ω is the set of possible networks, $P(G)$ and $P(\boldsymbol{\theta}|G)$ are prior probabilities of the network G and the parameter $\theta\boldsymbol{\theta}$, respectively. By using the density or probability functions, the posterior probability can be expressed as

$$\pi(G|\boldsymbol{X}) \propto \pi(G) \int \prod_{i=1}^{n} f(x_{i1}, \ldots, x_{ip}; \boldsymbol{\theta}_G) \pi(\boldsymbol{\theta}_G) d\boldsymbol{\theta}_G \quad (4)$$

BDe metric

Note that since the form of $\boldsymbol{\theta}$ is equivalent to the network structure, we write $\boldsymbol{\theta}_G$ as a parameter vector given network G. The problem now is how to compute the high-dimensional integration in equation (4). Usually this integration can be solved analytically by using the conjugate prior as $\pi(\boldsymbol{\theta}_G)$.

Conjugate prior

Discrete model

Laplace approximation

For the discrete model defined by equation (3), the parameter vector $\boldsymbol{\theta}_G = (\boldsymbol{\theta}_1^{\mathrm{T}}, \ldots, \boldsymbol{\theta}_p^{\mathrm{T}})^{\mathrm{T}}$ can be rewritten as $\boldsymbol{\theta}_j^{\mathrm{T}} = (\theta_{j11}, \ldots, \theta_{jQ_j m})^{\mathrm{T}}$, where θ_{jkl} corresponds to $P(X_{ij} = u_l | \boldsymbol{P}_{i-1,j} = \boldsymbol{u}_{jk})$. In this case, the Dirichlet distribution is often used as the prior distribution on the parameter θ_{jkl}:

Dirichlet distribution

Prior distribution on the parameter

$$D(\boldsymbol{\theta}_j|\boldsymbol{\alpha}_j) = \frac{\Gamma\left(\sum_{k'}\sum_{l'} \alpha_{jk'l'}\right)}{\prod_{k'}\prod_{l'} (\Gamma(\alpha_{jk'l'}))} \prod_k \prod_l \theta_{jkl}^{\alpha_{jkl}-1}$$

where $\Gamma(\cdot)$ is the gamma function and $\boldsymbol{\alpha}_j = (\alpha_{j11}, \ldots, \alpha_{jQ_j m})^{\mathrm{T}}$ is a hyperparameter vector in the Dirichlet distribution.

Then the integration in the marginal likelihood can be solved in a closed form:

$$\int f(x_{11}, \ldots, x_{np}; \boldsymbol{\theta}_G) \pi(\boldsymbol{\theta}_G|\boldsymbol{\alpha}) \mathrm{d}\boldsymbol{\theta}_G = \prod_{j=1}^{p} \prod_{k=1}^{Q_j} \frac{\Gamma\left(\sum_l \alpha_{\mathrm{jkl}}\right)}{\Gamma\left(\sum_l \alpha_{jkl} + N_{jkl}\right)} \prod_{l=1}^{m} \frac{\Gamma(\alpha_{jkl} + N_{jkl})}{\Gamma(\alpha_{jkl})}$$

where $f(x_{11}, \ldots x_{np}; \theta_G)$ is defined by equation (3) and $\pi(\theta_G|\alpha) = \Pi_j D(\theta_j|\alpha_j)$ with $\alpha = \alpha_1^{\mathrm{T}}, \ldots, \alpha_p^{\mathrm{T}})^{\mathrm{T}}$.

In Heckerman *et al.*,[9] when $\sum_k \sum_l \alpha_{jkl}$ is assumed to be constant, the posterior probability of the network results in the BDe metric. Note that Friedman *et al.*[8] and Smith *et al.*[10] used the BDe metric as a criterion for learning networks.

Continuous model

For computing high-dimensional integration in the marginal likelihood, Kim *et al.*[13] used Laplace approximation.[17,18] An advantage of using the Laplace approximation is that it is not necessary to consider the use of the conjugate prior distribution. Let $\pi(\theta_G|\lambda)$ be a prior distribution on $\boldsymbol{\theta}_G$ with a hyperparameter vector λ, satisfying $\log \pi(\theta_G|\lambda) = O(n)$. By using Laplace approximation, the integration can be computed as:

$$\int f(x_{11}, \ldots, x_{np}; \boldsymbol{\theta}_G)\pi(\boldsymbol{\theta}_G|\boldsymbol{\lambda})d\boldsymbol{\theta}_G$$
$$= \int \exp\{nl_\lambda(\boldsymbol{\theta}_G|\boldsymbol{X})\}d\boldsymbol{\theta}_G$$
$$= \frac{(2\pi/n)^{r/2}}{|J_\lambda(\hat{\boldsymbol{\theta}}_G)|^{1/2}} \exp\left\{nl_\lambda(\hat{\boldsymbol{\theta}}_G|\boldsymbol{X})\right\} \left\{1 + O_p(n^{-1})\right\}$$

where r is the dimension of q_G,

$$l_\lambda(\boldsymbol{\theta}_G|\boldsymbol{X}) = \log f(x_{11}, \ldots, x_{np}; \boldsymbol{\theta}_G)/n$$
$$= \log \pi(\boldsymbol{\theta}_G|\lambda)/n$$
$$J_\lambda(\boldsymbol{\theta}_G) = -\partial^2\{l_\lambda(\boldsymbol{\theta}_G|\boldsymbol{X})\}/\partial\boldsymbol{\theta}_G\partial\boldsymbol{\theta}_G^{\mathrm{T}}$$

and $\hat{\boldsymbol{\theta}}_G$ is the mode of $l_\lambda(\boldsymbol{\theta}_G|\boldsymbol{X})$.

BNRC dynamic

Then Kim *et al.*[13] defined a criterion, called $\mathrm{BNRC_{dynamic}}$, of the form:

$$BNRC_{\mathrm{dynamic}}(G) = -2\log \left\{\pi(G)\int f(x_{11}, \ldots, x_{np}; \boldsymbol{\theta}_G)\pi(\boldsymbol{\theta}_G|\boldsymbol{\lambda})d\boldsymbol{\theta}_G\right\}$$
$$\approx -2\log\pi(G) - r\log(2\pi/n) + \log|J_\lambda(\hat{\boldsymbol{\theta}}_G)| - 2nl_\lambda(\hat{\boldsymbol{\theta}}_G|\mathbf{X})$$

The optimal network is chosen such that the criterion $\mathrm{BNRC_{dynamic}}$ is minimal.

Prior probability of a network

MDL encoding

For computing the score of criterion (4), we need to consider a prior probability of a network denoted by $\pi(G)$. Friedman and Goldszmidt[19] used a prior based on the MDL encoding of network G. Kim *et al.*[13] set $\pi(G)$ based on the number of parent genes.

Biological knowledge

On the other hand, we can embed biological knowledge in a prior probability. Imoto *et al.*[20] constructed a prior probability of a network based on biological knowledge such as binding site information, DNA–protein interaction and so on.

COMPUTATIONAL EXPERIMENT

In this section, *S. cerevisiae* cell cycle time series microarray data[14] are analysed. The DBN and non-parametric regression model of Kim *et al.*[13] are applied to the data. These data contain two short time series (two time points; cln3, clb2) and four medium length time series (18, 24, 17 and 14 time points; alpha, cdc15, cdc28 and elu). In the estimation of a gene network, we use the four medium length time series. The first observation of the target gene and the last observation of parent genes are ignored, for each time series.

First, we focus on the cell cycle pathway compiled in the KEGG database.[21] The target network is around CDC28 (YBR160w; cyclin-dependent protein kinase). This network contains 45 genes and the partial pathway registered in KEGG is shown in Figure 3(a). Figures 3(b) and (c) are the resulting networks of the BN model[4,5] and the DBN model[13] respectively. A shaded circle represents the genes that compose a complex. The edges inside these circles are considered as correct edges since genes inside the same circle will co-express with some delay. A correct estimation is indicated by an edge attached with a circle. A triangle represents either a misdirected edge or an edge skipping at most one gene. A cross represents a wrong estimation.

Our second example is the metabolic pathway reported by DeRisi *et al.*[22] This network contains 57 genes and the target pathway is partially shown in Figure 4(a). Compared with the BN and non-parametric regression, the number of false positives in the DBN model shown in Figures 3(c) and 4(c) is much smaller than those in Figures 3(b) and 4(b).

CONCLUSION

A general framework for the DBN models for constructing gene networks from time series microarray data is summarised. Both discrete and continuous models are shown in detail and criteria were introduced for evaluating these models. Three discrete models[8,10–12] and one continuous model[13] were focused on and the strengths and weaknesses of these methods were discussed.

We need to find the optimal network

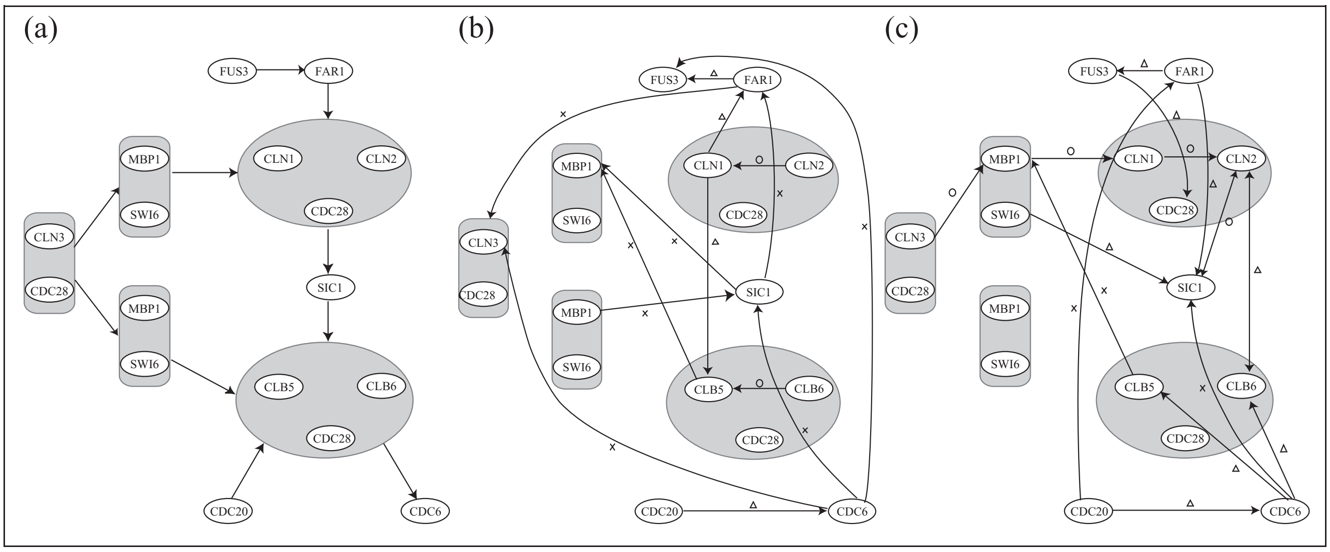

Figure 3: Cell cycle pathway compiled in KEGG: (a) target pathway; (b) result of the BN model and (c) result of the DBN model

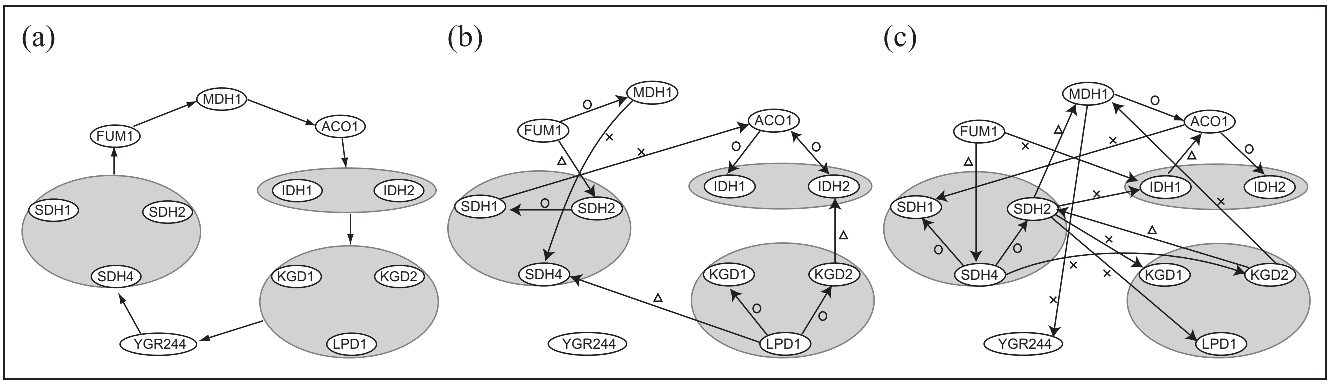

Figure 4: Metabolic pathway reported by DeRisi *et al.*:[22] (a) target pathway; (b) result of the BN model and (c) result of the DBN model

Heuristic search method

that gives the best score. The number of possible DAGs however is huge, even if we estimate a network containing a somewhat smaller number of genes. For example, when we have 20 genes, the number of DAGs is over 10^{72}. Therefore the use of heuristic search methods is required and several methods such as greedy hill-climbing,[8,13] simulated annealing[10,23] and junction tree algorithm[12] have been used to find a solution. Development of effective methods for learning networks is needed to find a better solution.

Although microarray data gives us valuable information, it is difficult to know whole gene networks by using only microarray data. Like Ong *et al.*[12] and Imoto *et al.*,[20] many researchers are now interested in combining microarray data with another technique, such as protein–protein interactions and binding site information,[24] for extracting more information.

Acknowledgment

We would like to thank the anonymous referees for valuable and helpful comments.

References

1. Akutsu, T., Miyano, S. and Kuhara, S. (1999), 'Identification of genetic networks from a small number of gene expression patterns under the Boolean network model', in 'Pacific

Symposium on Biocomputing', World Scientific, Singapore, Vol. 4, pp. 17–28.

2. Akutsu, T., Miyano, S. and Kuhara, S. (2000), 'Inferring qualitative relations in genetic networks and metabolic pathways', *Bioinformatics*, Vol. 16, pp. 727–734.
3. Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000), 'Using BN to analyze expression data', *J. Comp. Biol.*, Vol. 7, pp. 601–620.
4. Imoto, S., Goto, T. and Miyano, S. (2002), 'Estimation of genetic networks and functional structures between genes by using BN and nonparametric regression', in 'Pacific Symposium on Biocomputing', World Scientific, Singapore, Vol. 7, pp. 175–186.
5. Imoto, S., Kim, S., Goto, T., Aburatani, S. *et al.* (2003), 'BN and nonparametric heteroscedastic regression for nonlinear modeling of genetic network', *J. Bioinformatics and Computational Biology*, in press. (Preliminary version has appeared in *Proc. 1st IEEE Computer Society Bioinformatics Conference*, pp. 219–227, 2002.)
6. Chen, T., He, H. L. and Church, G. M. (1999), 'Modeling gene expression with differential equations', *Pacific Symposium on Biocomputing*, World Scientific, Singapore, Vol. 4, pp. 29–40.
7. De Hoon, M. J. L., Imoto, S., Kobayashi, K. *et al.* (2003), 'Inferring gene regulatory networks from time-ordered gene expression data of *Bacillus subtilis* using differential equations', in 'Pacific Symposium on Biocomputing', World Scientific, Singapore, Vol. 8, pp. 17–28.
8. Friedman, N., Murphy, K. and Russell, S. (1998), 'Learning the structure of dynamic probabilistic networks', in 'Proceedings of the 14th Conference on the Uncertainty in Artificial Intelligence', Morgan Kaufmann, San Mateo, CA, pp. 139–147.
9. Heckerman, D., Geiger, D. and Chickering, D. M. (1995), 'Learning BNs: the combination of knowledge and statistical data', *Machine Learning*, Vol. 20, pp. 197–243.
10. Smith, V. A., Jarvis, E. D. and Hartemink, A. J. (2002), 'Evaluating functional network inference using simulations of complex biological systems', *Bioinformatics*, Vol. 18 (ISMB2002), pp. S216–S224.
11. Smith, V. A., Jarvis, E. D. and Hartemink, A. J. (2003), 'Influence of network topology and data collection on network inference', in 'Pacific Symposium on Biocomputing', World Scientific, Singapore, Vol. 8, pp. 164–175.
12. Ong, I. M., Glasner, J. D. and Page, D. (2002), 'Modelling regulatory pathways in *E. coli* from time series expression profiles', *Bioinformatics*, Vol. 18 (ISMB2002), pp. S241–S248.
13. Kim, S., Imoto, S. and Miyano, S. (2003), 'DBN and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data', *Proc. Computational Methods in Systems Biology*, Lecture Note in Computer Science, Vol. 2602, Springer-Verlag, Berlin, Germany, pp. 104–113.
14. Spellman, P. T., Sherlock, G., Zhang, M. Q. *et al.* (1998), 'Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization', *Mol. Biol. Cell*, Vol. 9, pp. 3273–3297.
15. Friedman, N. and Goldszmidt, M. (1996), 'Discretizing continuous attributes while learning BNs', 'Proceedings of the 13th International Conference on Machine Learning', Morgan Kaufmann, San Mateo, CA, pp. 157–165.
16. Pe'er, D., Regev, A., Elidan, G. and Friedman, N. (2001), 'Inferring subnetworks from perturbed expression profiles', *Bioinformatics*, Vol. 17 (ISMB2001), pp. S215–S224.
17. Davison, A. C. (1986), 'Approximate predictive likelihood', *Biometrika*, Vol. 73, pp. 323–332.
18. Tinerey, L. and Kadane, J. B. (1986), 'Accurate approximations for posterior moments and marginal densities', *J. Amer. Statist. Assoc.*, Vol. 81, pp. 82–86.
19. Friedman, N. and Goldszmidt, M. (1998), 'Learning BNs with local structure', in Jordan, M. I. (Ed.), 'Learning in Graphical Models', Kluwer Academic, USA, pp. 421–459.
20. Imoto, S., Higuchi, T., Goto, T. *et al.* (2003), 'Combining microarrays and biological knowledge for estimating gene networks via BNs', in 'Proceedings of the 2nd IEEE Computer Society Bioinformatics Conference', in press.
21. URL: http://www.genome.ad.jp/kegg/
22. DeRisi, J., Lyer, V. R. and Brown, P. O. (1997), 'Exploring the metabolic and gene control of gene expression on a genomic scale', *Science*, Vol. 278, pp. 680–686.
23. Hartemink, A. J., Gifford, D. K., Jaakkola, T. S. and Young, R. A. (2002), 'Combining location and expression data for principled discovery of genetic regulatory networks', in 'Pacific Symposium on Biocomputing', World Scientific, Singapore, Vol. 7, pp. 437–449.
24. Tamada, Y., Kim, S., Bannai, H. *et al.* (2003), 'Estimating gene networks from gene expression data by combining BN model with promoter element detection', *Bioinformatics*, Vol. 19 (ECCB2003), in press.