Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Statistical Methodology 3 (2006) 1–16

Statistical Methodology
Official Journal of the International Indian Statistical Association

www.elsevier.com/locate/stamet

Error tolerant model for incorporating biological knowledge with expression data in estimating gene networks

Seiya Imoto[a,*,1], Tomoyuki Higuchi[b,1], Takao Goto[a], Satoru Miyano[a]

[a] *Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan*

[b] *Institute of Statistical Mathematics, 4-6-7, Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan*

Received 28 February 2005; received in revised form 16 September 2005; accepted 17 September 2005

Abstract

We propose a novel statistical method for estimating gene networks based on microarray gene expression data together with information from biological knowledge databases. Although a large amount of gene regulation information has already been stored in some biological databases, there are still errors and missing facts due to experimental problems and human errors. Therefore, we cannot blindly use them for understanding gene regulation and a robust procedure with a statistical model for using such database information is required. By using gene expression data, we provide a probabilistic framework of a joint learning model for repairing database information and for estimating a gene network based on dynamic Bayesian networks, simultaneously. To show the effectiveness of the proposed method, we analyze *Saccharomyces cerevisiae* cell-cycle gene expression data together with KEGG information.

Keywords: Microarray data; Biological knowledge; Error tolerant model; Gene network; Bayesian network

1. Introduction

In recent years, a lot of attention has been focused on combining microarray gene expression data and other types of genomic data for estimating gene networks [3,8,17,21,29,33,34,38]. Various types of genomic data, such as gene expression, protein–protein interactions, protein–DNA

[*] Corresponding author. Tel.: +81 3 5449 5615; fax: +81 3 5449 5442.

E-mail addresses: imoto@ims.u-tokyo.ac.jp (S. Imoto), higuchi@ism.ac.jp (T. Higuchi), takao@ims.u-tokyo.ac.jp (T. Goto), miyano@ims.u-tokyo.ac.jp (S. Miyano).

[1] These authors contributed equally to this work.

1572-3127/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.stamet.2005.09.013

interactions and binding site information, have been observed systematically. Many relationships among genes are then collected based on these data and stored in biological databases. However, due to experimental problems and human errors, databases are still incomplete and incorrect. We therefore cannot blindly use them for understanding gene regulatory mechanisms. Like expression data, the information in biological databases should therefore be considered as observational data that contain noise. Hence, development of statistical methods for extracting reliable information from such noisy genomic data is considered to be an important problem in bioinformatics.

Various computational methods have been proposed for extracting gene regulation information from gene expression data [40], such as Boolean networks [1,27,35,36], ordinary differential equations [4,7] and Bayesian networks [12,13,16,18,19,30]. Among them, Bayesian networks provide a useful probabilistic framework for extracting causal relationships from high-dimensional noisy data. Imoto et al. [21] proposed a general framework for estimating gene networks by using microarray gene expression data together with biological knowledge via Bayesian networks. Database information is modeled as a Bayesian prior probability of the graph and hyperparameters included in the prior probability control the balance between information on gene expression data and biological knowledge.

Although Imoto et al. [21] succeeded in extracting more reliable information than the previous methods [18,19] that are based on gene expression data only, a problem that still remains to be solved is how we treat errors and missing facts in a biological knowledge database. To solve this problem, in this paper, we propose an error tolerant model for incorporating biological knowledge with expression data in estimating gene networks. The purpose of this paper is realized by using a self-repairing system for biological knowledge databases based on information from gene expression data. Our method can repair database information based on the proposed statistical model and simultaneously estimate a gene network via Bayesian networks. The proposed method can be easily extended to the dynamic Bayesian networks. The dynamic Bayesian networks are an extension of the Bayesian networks to find even cyclic causal relations among random variables based on time series data.

A simple way to repair database information is to compare the initial database information with the network estimated from the expression data and the initial database information. However, it is possible that the estimated network is affected by errors and missing facts of the database and simple updating possibly leads to overfitting or overlearning to the expression data. On the other hand, the proposed method repairs the database information based on the statistical model and re-estimates a gene network based on the updated database information and expression data.

The proposed method has two aims depending on the quality of the database information: for high-quality database information, since it is unnatural to revise database information based on microarray data, the proposed method can automatically add missing information to the database. On the other hand, for low-quality biological information such as hypotheses, we can test information of such databases based on the proposed method. To show the effectiveness of the proposed method, we analyze *Saccharomyces cerevisiae* cell-cycle gene expression data collected by Spellman et al. [37] together with information of KEGG database [22] as a real data example.

2. Error tolerant model

2.1. Probabilistic framework

In this section, we introduce a probabilistic framework for the proposed error tolerant model that includes a self-repairing biological knowledge database system. Suppose that $\boldsymbol{X}_n$ is an $n \times p$

gene expression data matrix whose (i, j)th element is the expression value of gene$_j$ measured by ith microarray. For time series gene expression data, we denote the (i, j)th element of $\boldsymbol{X}_n$ by $x_j(t_i)$ so that it should explicitly indicate the time point t_i, satisfying $t_1 < \cdots < t_n$. Also, we assume that some information about the regulations among genes is initially known. We then summarize that information as a $p \times p$ matrix $\boldsymbol{A}_0 = (a^0_{ij})_{1\le i,j\le p}$ as follows: if we know that gene$_i$ regulates gene$_j$, we set $a^0_{ij} = 1$. On the other hand, if we know gene$_i$ does not regulate gene$_j$, we set $a^0_{ij} = 2$. In addition, we set $a^0_{ij} = 0$ for edges that are not stored in the database. Note that the initial database information $\boldsymbol{A}_0$ happens to contain some errors and missing facts. Also, the negative data such as gene$_i$ does not regulate gene$_j$, i.e. $a^0_{ij} = 2$ usually are not kept in the database. However, we may use the information of sub-cellular localization to create a negative data [15].

Our aim is to find the optimal graph $\hat{G}$ and the optimal updated database information $\hat{\boldsymbol{A}}$ that maximize the conditional joint probability

$$P(G, \boldsymbol{A}|\boldsymbol{X}_n, \boldsymbol{A}_0), \tag{1}$$

where the (i, j)th element of $\boldsymbol{A}$, a_{ij}, is the updated information from a^0_{ij}. The conditional joint probability (1) is then rewritten as

$$P(G, \boldsymbol{A}|\boldsymbol{X}_n, \boldsymbol{A}_0) = \frac{P(\boldsymbol{X}_n, G, \boldsymbol{A}|\boldsymbol{A}_0)}{P(\boldsymbol{X}_n|\boldsymbol{A}_0)},$$

where $P(\boldsymbol{X}_n|\boldsymbol{A}_0) = \sum_G \sum_{\boldsymbol{A}} P(G, \boldsymbol{A}, \boldsymbol{X}_n|\boldsymbol{A}_0)$ is the normalizing constant and does not relate to the selection of G and $\boldsymbol{A}$. Therefore, given $\boldsymbol{X}_n$, the maximization of the conditional joint probability (1) is equivalent to the maximization of $P(\boldsymbol{X}_n, G, \boldsymbol{A}|\boldsymbol{A}_0)$. When the database information $\boldsymbol{A}_0$ is given, the conditional joint probability $P(\boldsymbol{X}_n, G, \boldsymbol{A}|\boldsymbol{A}_0)$ is then decomposed as

$$P(\boldsymbol{X}_n, G, \boldsymbol{A}|\boldsymbol{A}_0) = P(\boldsymbol{X}_n|G)P(G|\boldsymbol{A})P(\boldsymbol{A}|\boldsymbol{A}_0). \tag{2}$$

Here, $P(\boldsymbol{X}_n|G, \boldsymbol{A}) = P(\boldsymbol{X}_n|G)$ holds in our model. We should note that we could estimate G based on $P(G|\boldsymbol{X}_n, \boldsymbol{A}_0) \propto P(\boldsymbol{X}_n|G)P(G|\boldsymbol{A}_0)$, where $\boldsymbol{A}$ is summed out. However, this modeling is sometimes infeasible in practice, because the computational complexity of this marginalization is $O(3^{p^2})$ for simple enumeration. In addition, our interest is not only in the estimation of G, but also in the update of A_0. In the following sections, we describe statistical models for representing $P(\boldsymbol{X}_n|G)$, $P(G|\boldsymbol{A})$ and $P(\boldsymbol{A}|\boldsymbol{A}_0)$.

2.2. Bayesian networks

In the context of Bayesian networks, a gene is regarded as a random variable and the gene network is modeled as a directed acyclic graph with the first-order Markov relationships between genes. Let $X_1, \ldots, X_p$ be random variables corresponding to genes. Using the above assumptions, based on the structure of the directed acyclic graph, the joint probability of all genes can be decomposed as

$$P(X_1, \ldots, X_p) = \prod_{j=1}^{p} P(X_j|\boldsymbol{P}_j),$$

where $\boldsymbol{P}_j$ is a random variable vector of direct parent genes of gene$_j$. For example, if gene$_2$ and gene$_3$ are the direct parents of gene$_1$, we have $\boldsymbol{P}_1 = (X_2, X_3)^{\mathrm{T}}$. Here, $\boldsymbol{a}^{\mathrm{T}}$ is the transpose of the vector $\boldsymbol{a}$.

For time series gene expression data described in the previous section, we employ dynamic Bayesian network models [14,28] for computing $P(\boldsymbol{X}_n|G)$. The dynamic Bayesian network assumes that the states of genes at time t_i depend only on those at time t_{i-1}, and the relationships between genes are stable at any time points. Let $X_j(t_i)$ be a random variable corresponding to gene$_j$ at time t_i. Using the above assumptions, the joint probability of all random variables can be decomposed as

$$P(X_1(t_1), \dots, X_p(t_n)) = P(X_1(t_1), \dots, X_p(t_1)) \prod_{j=1}^{p} \prod_{i=2}^{n} P(X_j(t_i)|\boldsymbol{P}_j(t_{i-1})),$$

where $\boldsymbol{P}_j(t_i)$ is a random variable vector of direct parent genes of gene$_j$ at time t_i. For example, if gene$_2$ and gene$_3$ are the direct parents of gene$_1$, we have $\boldsymbol{P}_1(t_i) = (X_2(t_i), X_3(t_i))^{\mathrm{T}}$.

In the context of dynamic Bayesian networks, since the expression data take continuous variables, the likelihood of the expression data $\boldsymbol{X}_n$ for a given graph structure G is expressed by densities of the form

$$f(\boldsymbol{X}_n|\boldsymbol{\theta}, G) = f_0(\boldsymbol{x}(t_1)|\boldsymbol{\theta}_0) \prod_{j=1}^{p} \prod_{i=2}^{n} f_j(x_j(t_i)|\boldsymbol{p}_j(t_{i-1}), \boldsymbol{\theta}_j), \tag{3}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_0^{\mathrm{T}}, \dots, \boldsymbol{\theta}_p^{\mathrm{T}})^{\mathrm{T}}$ is the parameter vector and $\boldsymbol{p}_j(t_i)$ is the gene expression value vector of the parents of gene$_j$ at time t_i and $\boldsymbol{x}(t_1)$ is the gene expression vector at the first time point. For computing $P(\boldsymbol{X}_n|G)$, we take the marginal likelihood that is given by integrating the joint density of $\boldsymbol{X}_n$ and $\boldsymbol{\theta}$ over the parameter $\boldsymbol{\theta}$:

$$P(\boldsymbol{X}_n|G) = \int f(\boldsymbol{X}_n, \boldsymbol{\theta}|G)\, \mathrm{d}\boldsymbol{\theta} = \int f(\boldsymbol{X}_n|\boldsymbol{\theta}, G)\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})\, \mathrm{d}\boldsymbol{\theta}, \tag{4}$$

where $\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})$ is a prior distribution on the parameter $\boldsymbol{\theta}$, and $\boldsymbol{\lambda}$ is the hyperparameter vector that specifies the shape of $\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})$. In our modeling, the hyperparameter works as a smoothing parameter in a nonparametric regression model that controls the amount of smoothness of the fitted curve. In addition, we can optimize the number of B-splines by using information criteria. However, in this paper, we use 20 B-splines for constructing each smooth function $m_{jk}(\cdot)$ and control the smoothness of the fitted curve by the hyperparameter in order to reduce computational time. This strategy is called P-splines [11].

Although the proposed method does not depend on the type of dynamic Bayesian network models, i.e. both discrete and continuous dynamic Bayesian networks can be used, we employ the dynamic Bayesian network and nonparametric regression model with B-splines [6,20] proposed by Kim et al. [24] (see also Kim et al. [23]). In our model, the relationship between a gene $x_j(t_i)$ and its parents $\boldsymbol{p}_j(t_{i-1}) = (p_{j1}(t_{i-1}), \dots, p_{jk_j}(t_{i-1}))^{\mathrm{T}}$ is represented by

$$x_j(t_i) = m_{j1}(p_{j1}(t_{i-1})) + \cdots + m_{jk_j}(p_{jk_j}(t_{i-1})) + \varepsilon_{ij},$$

where ε_{ij} depends independently on the normal distribution with mean 0 and variance σ_j^2 and $m_{jk}(\cdot)$ is a smooth function given by using B-splines

$$m_{jk}(x) = \sum_{m=1}^{M_{jk}} \gamma_{mk}^{(j)} b_{mk}^{(j)}(x), \qquad k = 1, \dots, k_j.$$

Here $\{b_{1k}^{(j)}(x), \dots, b_{M_{jk}k}^{(j)}(x)\}$ is the prescribed set of B-splines. Therefore, the conditional density $f_j(x_j(t_i)|\boldsymbol{p}_j(t_{i-1}), \boldsymbol{\theta}_j)$ can be expressed as

$$f_j(x_j(t_i)|\boldsymbol{p}_j(t_{i-1}), \boldsymbol{\theta}_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left[-\frac{\left\{ x_j(t_i) - \sum_{k,m} \gamma_{mk}^{(j)} b_{mk}^{(j)}(p_{jk}(t_{i-1})) \right\}^2}{2\sigma_j^2} \right],$$

where $\boldsymbol{\theta}_j$ contains $\gamma_{mk}^{(j)}$'s and σ_j^2.

Note that the dynamic Bayesian networks enable us to estimate the directed cycles in the network. Also, our model does not consider the intra-slice connections. Since there are several instantaneous correlations between genes, this assumption is somewhat strong. As long as the acyclicity holds in the intra-slice connections, we might construct a dynamic Bayesian network with intra-slice connections. In addition, a possible improvement of our Bayesian and dynamic Bayesian network models is to use hidden variables for representing unmeasurable products in RNA expression data such as protein concentrations, degradation, RNA interference and so on, which are possibly related to gene regulation.

2.3. *Prior probability of the graph*

For estimating gene networks, Imoto et al. [21] proposed the use of biological knowledge as a prior probability of the graph. According to Imoto et al. [21], in this section, we explain how we construct a prior probability of the graph $P(G|\boldsymbol{A})$ based on the database information $\boldsymbol{A}$. First, we allocate a value ζ_k to the edge from gene$_i$ to gene$_j$, if $a_{ij} = k$. That is, since a_{ij} takes one of three values, 0, 1 or 2, we use ζ_0, ζ_1 and ζ_2 for discriminating biological knowledge on each edge. In addition, we set $\zeta_0 = 0 < \zeta_1 < \zeta_2$. The prior probability of the graph G can be expressed as

$$\pi(G|\boldsymbol{A}) = Z^{-1} \exp\left\{ - \sum_{(i,j)\in G} \zeta_{a_{ij}} \right\},$$

where the sum $\sum_{(i,j)\in G}$ is taken over the existing edges in G and Z is the normalizing constant given by $Z = \sum_{G\in\mathcal{G}} \exp\{-\sum_{(i,j)\in G} \zeta_{a_{ij}}\}$. Here $\mathcal{G}$ is the set of possible directed graphs. It should be noticed that the values ζ_1 and ζ_2 are parameters that need to be optimized, see Imoto et al. [21]. We optimize the values of ζ_1 and ζ_2 by using the criterion, called BNRC$_{DB}$, defined in Section 2.5.

For computing the conditional joint probability (1), we need to calculate the normalizing constant Z. Although the computation of Z is intractable even for moderately sized gene networks based on Bayesian network models [21], we can compute the exact value of Z for the dynamic Bayesian network models. In the biological knowledge matrix $\boldsymbol{A}$, suppose that the numbers of 1's, 2's and 0's are z_1, z_2 and z_0, respectively. Note that $z_1 + z_2 + z_0 = p^2$ holds. Let us consider the situation that, in a graph G, the numbers of ζ_1, ζ_2 and ζ_0 edges are α, β and γ, respectively. The number of such graphs is then $\binom{z_1}{\alpha}\binom{z_2}{\beta}\binom{z_0}{\gamma}$. Thus the normalizing constant Z can be obtained by

$$Z = 2^{z_0} \sum_{\alpha=0}^{z_1} \sum_{\beta=0}^{z_2} \binom{z_1}{\alpha}\binom{z_2}{\beta} \exp(-\alpha\zeta_1 - \beta\zeta_2). \tag{5}$$

Here we use $\sum_{\gamma=0}^{z_0} \binom{z_0}{\gamma} = 2^{z_0}$. Note that, for the Bayesian network models, Imoto et al. [21] computed an upper and a lower bound of Z. In practice, the upper bound of Imoto et al. [21] works well.

2.4. Database information model

The conditional probability $P(\boldsymbol{A}|\boldsymbol{A}_0)$ represents the transition probability when we update the database information from $\boldsymbol{A}_0$ to $\boldsymbol{A}$. In this section, we elucidate a statistical model for $P(\boldsymbol{A}|\boldsymbol{A}_0)$ that is essential to realize a self-repairing system for biological knowledge database. First, we define the function $d(a)$ by

$$d(a) = \begin{cases} 1 & \text{for } a = 1 \text{ or } 2, \\ 0 & \text{for } a = 0. \end{cases}$$

Note that we assume that the edges whose database statuses are 1 or 2 have almost the same accuracy. The function $d(a)$ then categorizes the edges into two groups: one group includes the edges that are contained in the database, and the other group is composed of the edges that are not contained in the database. The transition probability $P\{d(a_{ij})|d(a_{ij}^0)\}$ is then constructed by using the Bernoulli distribution of the form

$$P\{d(a_{ij})|d(a_{ij}^0)\} = P\{d(a_{ij}^0)\}^{d(a_{ij})}[1 - P\{d(a_{ij}^0)\}]^{1-d(a_{ij})}.$$

Therefore, we model $P(\boldsymbol{A}|\boldsymbol{A}_0)$ by the product of the Bernoulli distributions

$$P(\boldsymbol{A}|\boldsymbol{A}_0) = \prod_{i=1}^{p}\prod_{j=1}^{p} P\{d(a_{ij}^0)\}^{d(a_{ij})}[1 - P\{d(a_{ij}^0)\}]^{1-d(a_{ij})}. \tag{6}$$

We set a high probability for $P\{d(a_{ij}^0) = 1\}$, because the information on ζ_1 or ζ_2 edges is rather reliable. In the last section, we set $P\{d(a_{ij}^0) = 1\} = p_h = 0.9$. On the other hand, since there is no information about ζ_0 edges in the database, we set $P\{d(a_{ij}^0) = 0\} = p_m = 0.5$. Note that we do not allow the transition from $a_{ij}^0 = 1$ to $a_{ij} = 2$ and vice versa.

If the edge from gene$_i$ to gene$_j$ is stored as a known relationship in the database, i.e. $a_{ij}^0 = 1$, but this edge is not observed from the gene expression data, we remove this edge from the database, i.e. we change $a_{ij}^0 = 1$ to $a_{ij} = 0$, if it leads to an increase in the conditional joint probability (2). Also, if the edge from gene$_k$ to gene$_l$ is clearly observed by the expression data, but the database does not contain this edge, i.e. $a_{kl}^0 = 0$. We then add this edge to the database, i.e. we change $a_{kl}^0 = 0$ to $a_{kl} = 1$, if the conditional joint probability (2) increases. In the next section, we represent our greedy hill-climbing algorithm for the joint learning of the optimal graph $\hat{G}$ and simultaneously the optimal updated database information $\hat{\boldsymbol{A}}$.

2.5. Model learning

For learning the graph G and the database information $\boldsymbol{A}$ based on $\boldsymbol{X}_n$ and $\boldsymbol{A}_0$, we first define a criterion based on the joint probability (2). To construct a criterion, we need to compute the marginal likelihood of the data $P(\boldsymbol{X}_n|G)$ given in (4). The Laplace approximation [5,25,39] can solve this problem analytically. For a function $s(\boldsymbol{\theta}|\boldsymbol{x}_n)$ satisfying $s(\boldsymbol{\theta}|\boldsymbol{x}_n) = O(1)$, we have a formula

$$\int \exp\{ns(\boldsymbol{\theta}|\boldsymbol{x}_n)\}\,\mathrm{d}\boldsymbol{\theta} = \frac{(2\pi/n)^{r/2}}{|J(\hat{\boldsymbol{\theta}}|\boldsymbol{x}_n)|^{1/2}} \exp\{ns(\hat{\boldsymbol{\theta}}|\boldsymbol{x}_n)\}\{1 + O_p(n^{-1})\},$$

where $\boldsymbol{x}_n = (x_1, \ldots, x_n)^{\mathrm{T}}$, $J(\boldsymbol{\theta}|\boldsymbol{x}_n) = \partial^2 s(\boldsymbol{\theta}|\boldsymbol{x}_n)/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}$, r is the dimension of $\boldsymbol{\theta}$. Here $\hat{\boldsymbol{\theta}}$ is the mode of $s(\boldsymbol{\theta}|\boldsymbol{x}_n)$. Replacing $s(\boldsymbol{\theta}|\boldsymbol{x}_n)$ by $n^{-1}\{\log f(\boldsymbol{X}_n|\boldsymbol{\theta}, G) + \log \pi(\boldsymbol{\theta}|\boldsymbol{\lambda})\}$, we defined a criterion, called BNRC$_{DB}$, as the score function by taking minus twice the logarithm of the conditional joint probability (2)

$$\begin{aligned}\mathrm{BNRC}_{DB}(G, \boldsymbol{A}, \zeta_1, \zeta_2) &= 2\log Z + 2\sum_{(i,j)\in G} \zeta_{a_{ij}} - r\log(2\pi n^{-1}) \\ &\quad + \log|J_\lambda(\hat{\boldsymbol{\theta}}|\boldsymbol{X}_n)| - 2\{\log f(\boldsymbol{X}_n|\hat{\boldsymbol{\theta}}, G) + \log \pi(\hat{\boldsymbol{\theta}}|\boldsymbol{\lambda})\} \\ &\quad + \sum_{i=1}^{p}\sum_{j=1}^{p}[d(a_{ij})\log P\{d(a_{ij}^0)\} + \{1 - d(a_{ij})\}\log(1 - P\{d(a_{ij}^0)\})],\end{aligned}$$

where $J_\lambda(\boldsymbol{\theta}|\boldsymbol{X}_n) = n^{-1}\partial^2\{\log f(\boldsymbol{X}_n|\boldsymbol{\theta}, G) + \log\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})\}/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}$ and $\hat{\boldsymbol{\theta}}$ is the mode of $\log f(\boldsymbol{X}_n|\boldsymbol{\theta}, G) + \log\pi(\boldsymbol{\theta}|\boldsymbol{\lambda})$. The details of the computation of $\hat{\boldsymbol{\theta}}$ and $J_\lambda(\boldsymbol{\theta}|\boldsymbol{X}_n)$ are available in [19]. We then choose the optimal graph $\hat{G}$ and the optimal updated database information $\hat{\boldsymbol{A}}$ by minimizing BNRC$_{DB}$. In practice, the value of BNRC$_{DB}$ is computed as the sum of the local scores for each gene and its direct parents, the details of which are described in [18]. For finding $\hat{G}$ and $\hat{\boldsymbol{A}}$, we cannot perform an exhaustive search method due to the computational complexity of learning G and $\boldsymbol{A}$. We therefore use a greedy hill-climbing algorithm for finding $\hat{G}$ and $\hat{\boldsymbol{A}}$ heuristically, described as follows:

Initial Step

Step 1 Estimate the graph G based on $\boldsymbol{X}_n$ and the initial database information $\boldsymbol{A}_0$ by the greedy hill-climbing algorithm described below.

Step 2 Make a list of edges that do not agree with the initial database information $\boldsymbol{A}_0$. The edges in this list can be considered as candidates for update. We call this list the candidate edge list.

Learning Step

Step 3 For each edge (i, j) in the candidate edge list, we update a_{ij} and estimate G based on $\boldsymbol{X}_n$ and the updated database information $\boldsymbol{A}$.

Step 4 If the update that gives the smallest BNRC$_{DB}$ in Step 3 causes a reduction of BNRC$_{DB}$, we accept this change and go to Step 5. Otherwise, no operation is performed for $\boldsymbol{A}$ and we finish the learning.

Step 5 Update the candidate edge list based on the estimated graph G and the updated database information $\boldsymbol{A}$.

Step 6 Repeat from Step 3 to Step 5 until the learning finishes at Step 4.

In Step 1 and Step 3, we estimate the graph G based on $\boldsymbol{X}_n$ and given $\boldsymbol{A}$ by using the greedy hill-climbing algorithm as follows:

Step A Set the values ζ_1 and ζ_2.

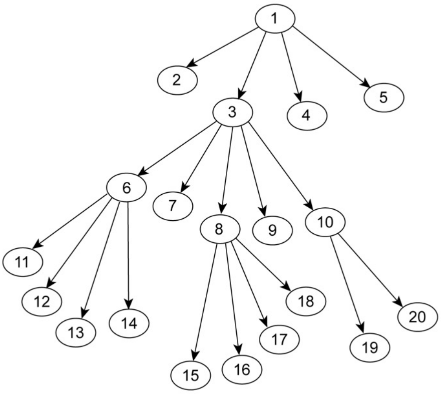
Step B Estimate G by minimizing BNRC$_{DB}$ under the given ζ_1 and ζ_2:

Step B-1 For each gene, either add or remove a parent gene, if it leads to a reduction in the criterion.

Step B-2 Repeat Step B-1 until the criterion reaches a minimum.

Step C Repeat Step A and Step B for the candidate values of ζ_1 and ζ_2.

Step D The optimal gene network is found from the candidate networks obtained in Step C.



(a)

$$g_1 = \varepsilon_1, \quad g_2 = .7g_1 + \varepsilon_2 \quad g_5 = .7g_1 + \varepsilon_5,$$

$$g_{10} = 1/\{1 + \exp(-4g_3)\} + \varepsilon_{10} \quad g_7 = 1.3g_7 + \varepsilon_7$$

$$g_3 = \begin{cases} -1 + \varepsilon_3 \ (g_1 \le -.5) \\ g_1 + \varepsilon_3 \ (|g_1| < .5) \\ 1 + \varepsilon_3 \ (g_1 \ge .5) \end{cases} \quad g_6 = \begin{cases} .8g_3 + \varepsilon_6 \ (g_3 \le -1) \\ (g_3 + 1)^{1.5} + \varepsilon_6 \ (-1 < |g_3| < 0) \\ 1 + \varepsilon_6 \ (g_3 \ge 1) \end{cases}$$

$$g_4 = \begin{cases} .4g_1 + 1 + \varepsilon_4 \ (|g_1| \ge .3) \\ (g_1 + 1)^2 + \varepsilon_4 \ (|g_1| < .3) \end{cases} \quad g_8 = \begin{cases} .2g_3 - 1 + \varepsilon_8 \ (g_3 \le .2) \\ 1.4g_3 + \varepsilon_8 \ (g_3 > .2) \end{cases}$$

$$g_{11} = .7g_6 + \varepsilon_{11}, \ g_{14} = .7g_6 + \varepsilon_{14}, \ g_{15} = 1/\{1 + \exp(-4g_8)\} + \varepsilon_{15}$$

$$g_9 = \begin{cases} .4g_3 + 1 + \varepsilon_9 \ (|g_3| \ge .3) \\ (g_3 + 1)^{1.2} + \varepsilon_9 \ (|g_3| < .3) \end{cases} \quad g_{13} = \begin{cases} .4g_6 + 1 + \varepsilon_{13} \ (|g_6| \ge .3) \\ (g_6 + 1)^2 + \varepsilon_{13} \ (|g_6| < .3) \end{cases}$$

$$g_{12} = \begin{cases} -1 + \varepsilon_{12} \ (g_6 < -.5) \\ g_6 + \varepsilon_{12} \ (|g_6| \le .5) \\ 1 + \varepsilon_{12} \ (g_6 > .5) \end{cases} \quad \begin{aligned} g_{16} &= .8g_8 + \varepsilon_{16} \\ g_{19} &= 1/\{1 + \exp(-4g_{10})\} + \varepsilon_{19} \\ g_{20} &= 1.1g_{10} + \varepsilon_{20} \end{aligned}$$

$$g_{17} = \begin{cases} .2g_8 - 1 + \varepsilon_{17} \ (g_8 \le .2) \\ 1.4g_8 + \varepsilon_{17} \ (g_8 > .2) \end{cases} \quad g_{18} = \begin{cases} .4g_8 + 1 \ (|g_8| > .3) \\ (g_8 + 1)^{1.2} \ (g_8 \le .3) \end{cases}$$

(b)

Fig. 1. True model for Monte Carlo simulations. (a) Artificial network. (b) Functional structures between nodes.

A gene network is re-estimated in Step 3 by using the candidate updated database information. Since the database information is updated, the prior probability of the graph, $\pi(G|\boldsymbol{A})$, changes. Therefore, it is possible that the optimal values of ζ_1 and ζ_2 change and we could obtain a different optimal graph compared to the optimal one in the previous step.

We note that since we use a greedy hill-climbing algorithm to learn G and $\boldsymbol{A}$, it cannot be guaranteed that the solutions $\hat{G}$ and $\hat{\boldsymbol{A}}$ are optimums. To find better solutions, we repeat the learning step described above 10 times and choose the best pair of $\hat{G}$ and $\hat{\boldsymbol{A}}$.

3. Computational experiments

3.1. Example using simulated data

Before we analyze real gene expression data, we conduct Monte Carlo simulations to examine the properties of the proposed method. We first set an artificial graph shown in Fig. 1(a) and the relationships between nodes listed in Fig. 1(b). The relationships between nodes in Fig. 1(b) reflect the saturations of gene expressions and some threshold expression values in gene regulations. We generate 100 observations, which correspond to 100 microarrays, from the true model. For the information of the biological knowledge database, we assume that we know the relationships between nodes described in Fig. 2(a). We remove three edges out of correct relationships and switch the direction of three other edges. That is, we consider the edges $(3, 6)$, $(3, 8)$ and $(3, 10)$ as missing information and the edges $(12, 6)$, $(15, 8)$ and $(19, 10)$ as errors in the database information. We apply the proposed method with Bayesian networks to those simulated data. Since the database information contains errors and missing relations, the purpose of the Monte Carlo simulations is not only to rebuild a gene network, but also to repair the database information.

Fig. 2(b) shows a typical example of the resulting networks in the Monte Carlo simulations. The black edges agree with the database information described in Fig. 2(a). The updated information is shown by blue edges. The term “revise” means that the proposed method correctly repaired the database information of the edge. The edges with “add” are added as new information to the database. The red edges are false positives (incorrectly estimated edges) and

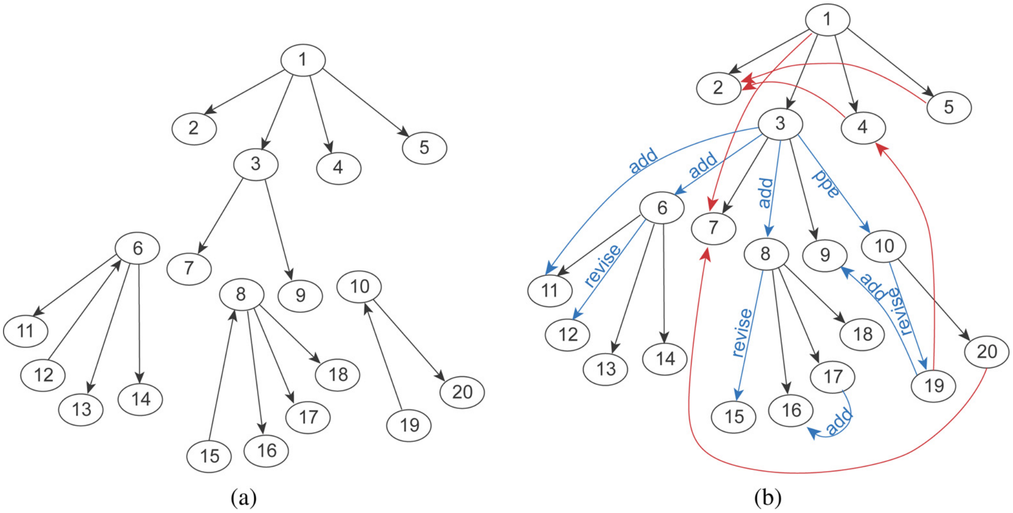


Fig. 2. (a) Database information denoted by $\boldsymbol{A}_0$. We remove three edges out of the correct relationships and switch the direction of three other edges. (b) An example of the resulting network of the Monte Carlo simulations. The black edges agree with the database information described in (a). The updated information is shown by blue edges. The red edges do not agree with the database information $\boldsymbol{A}_0$ and are not updated.

are not updated. Since the edges (12, 6), (15, 8) and (19, 10) are set as the opposite direction in the database, we can remove this information from the database, i.e. $a^0_{12,6} = a^0_{15,8} = a^0_{19,10} = 1$ are updated to $a_{12,6} = a_{15,8} = a_{19,10} = 0$, and estimate the edges of correct direction by using expression data. Also, the proposed method added the edges (3, 6), (3, 8) and (3, 10), which are missing in the initial database, as new information. However, from Fig. 2(b), the proposed method added some false positive edges to the database. In addition, we observed that the proposed method sometimes adds not only correct causal relationships but also false positive relationships. However, these falsely added relationships could be removed from the database by using other sets of microarray data or using the bootstrap method [9,10]. That is, incorrectly updated information may be fixed when the proposed method is applied to other microarray data. In fact, we observed that almost all false positive updates are removed by using other sets of simulated microarray data.

We repeat the Monte Carlo simulation 1000 times, that is we first generate 1000 datasets and then estimate $\hat{G}$ and $\hat{\boldsymbol{A}}$ for each dataset. Table 1 shows the edges that are updated more than 100 times out of 1000 repetitions. Fig. 3(a) and (b) show the distribution of the numbers of updates for all 380 possible edges. Note that the graph can be represented by the adjacent matrix and the number of non-diagonal elements is 380 for 20 node network. The edges with asterisks are correctly updated edges. From Table 1, we observe that the operation “add” is successfully done with high probability, but there are several false positives. However, the numbers of false positives, except for $g_{19} \rightarrow g_9$, are not large and may be acceptable. The reason why the number of additions of $g_{19} \rightarrow g_9$ becomes large is the setting of functions between nodes. In our setting, it is possible that an untrue relationship between g_{19} and g_9 is observed. On the other hand, the coverage of the operation “remove” is not high, but the number of false positives is very small. This feature is very natural in our situation, because (1) there is much missing information in the database; we can add information to the database in a positive way, but (2) the database information is created based on various kinds of knowledge including protein–protein

Table 1
The results of update edges in Monte Carlo simulations

Added edges	#	**Removed edges**	#
* $g_3 \rightarrow g_{10}$	875	* $g_{12} \rightarrow g_6$	207
* $g_3 \rightarrow g_6$	826	* $g_{19} \rightarrow g_{10}$	202
* $g_3 \rightarrow g_8$	530	* $g_{15} \rightarrow g_8$	194
$g_{19} \rightarrow g_9$	456		
$g_{10} \rightarrow g_9$	205		
$g_{19} \rightarrow g_4$	185		
$g_{10} \rightarrow g_4$	143		
$g_{15} \rightarrow g_{18}$	129		
$g_{10} \rightarrow g_8$	122		
$g_3 \rightarrow g_4$	112		

This table shows the edges that were updated more than 100 times out of 1000 Monte Carlo experiments. The **Added edges** are updated from $a_{ij}^0 = 0$ to $\hat{a}_{ij} = 1$, and the **Removed edges** are from $a_{ij}^0 = 1$ to $\hat{a}_{ij} = 0$. The columns "#" indicate the numbers of updates in 1000 Monte Carlo experiments.

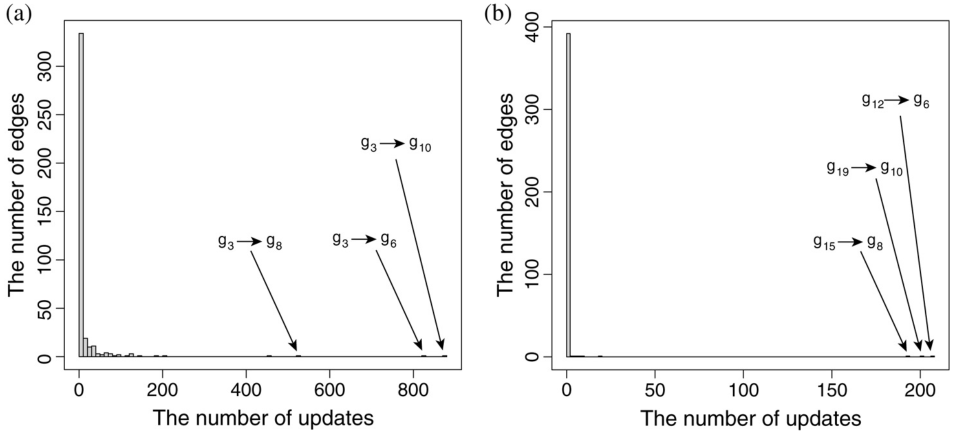


Fig. 3. The distributions of the numbers of updated edges for 380 possible edges. (a) The distribution of the numbers of updates with respect to the added edges. (b) The distribution of the numbers of updates with respect to the removed edges.

interactions. We thus need to pay much more attention when we remove information from the database. In this sense, the proposed method can perform reasonable database updates.

The results of the Monte Carlo simulations are summarized as follows: (1) In the case that there are the opposite direction edges against the true relationships in the database information: if these wrong relationships are nonlinear, we observed that the proposed method can revise this information by using gene expression data. We presume one of the reasons is that if expression data show nonlinear relationships, expression data might contain the information about the direction. On the other hand, if the relationships of the error edges are almost linear, it is difficult to decide the causal direction. However, the directed acyclic graph structure of the Bayesian networks could help to decide the causal direction. That is, in the Bayesian networks, the causal direction of an edge with a linear dependency may be found be considering edges in the neighborhood and the acyclicity condition. (2) In the case that some information is missing from the database: as shown in the example above, we can add such information to the database.

However, we need to be concerned about false positive updates. We described two ways to identify the false positive updates above. (3) In the case that we cannot find edges using gene expression data, but these are stored in the database: although we can remove such information from the database, we need to do this operation very carefully in practice. Gene expression data consider regulation of the transcriptional level only, while the database information may be broader. As the conclusion of the Monte Carlo simulations, we observed that the proposed method works well in practice.

3.2. Example using experimental data

As a real data application, we analyze *S. cerevisiae* cell-cycle gene expression data collected by Spellman et al. [37]. These data contain 77 microarrays and consist of two short time-courses (cln3, clb2: two time points) and four time-courses (alpha, cdc15, cdc28 and elu: 18, 24, 17 and 14 time points, respectively). Therefore, we use the dynamic Bayesian network model with B-spline nonparametric regression described in Section 2.2. We focus on 29 genes that are stored in KEGG [22] to be a cell-cycle subnetwork related to G1 and S phases. The KEGG database is one of the most well-established databases and the cell-cycle network in *S. cerevisiae* has been investigated for a long time in biology. Also, for estimating cell-cycle networks, we need to compile microarray data obtained by experiments affecting the cell-cycle related genes. In this sense, using Spellman's cell-cycle data with information of the KEGG database is suitable for estimating cell-cycle networks and for evaluating the proposed method. The database information matrix $\boldsymbol{A}_0$ is constructed by transforming the cell-cycle network in KEGG as follows: we assign the values ζ_1 and ζ_0 to the edges that are shown and not shown in the cell-cycle network in KEGG. Also, for the value ζ_2, we could regard the edges that show opposite direction of ζ_1 edges as known non-regulations and assign ζ_2 to them. Of course, all edges that show opposite direction of ζ_1 edges do not have enough biological evidence for known non-regulation edges. The proposed method, however, can allow such errors in the database information. It is possible that those errors are repaired by using expression data. As we noted in Section 2.1, it is possible that we use sub-cellular localization as a reliable source for $a_{ij} = 2$. In addition, there are many protein complexes in the cell-cycle network in KEGG, such as "*CLN1–CDC28*". We set ζ_1 for both directions, "*CLN1* → *CDC28*" and "*CLN1* ← *CDC28*". Note that, in this analysis, we omit protein complexes, ORC, MCM, APC and SCF, that consist of many subunits. A method for treating such protein complexes is described in [29].

In the estimation of a gene network, we use four time series data (alpha, cdc15, cdc28 and elu) and estimate a gene network from each time series data. Fig. 4(a) shows the resulting network obtained by superposing four estimated networks. We remove edges that are found from only one estimated network from Fig. 4(a). The black and red edges agree and do not agree with the information of the cell-cycle network in KEGG, respectively. The blue edges also do not agree with the KEGG information but are added as new information by the proposed method. The updated edges are listed in Table 2. Since we use four time series data, the column "Data" in Table 2 shows which time series data are used to update. Since the gene networks are generally condition specific, it is possible that the estimated networks from different sets of microarray data are not consistent. Therefore, we show edges that appear in two or more estimated networks in order to extract more reliable information. It should be noted that there are important edges that appear in only one estimated network. Therefore, we also need to pay attention to each estimated network. By removing the red edges from Fig. 4(a), Fig. 4(b) shows a gene network with the

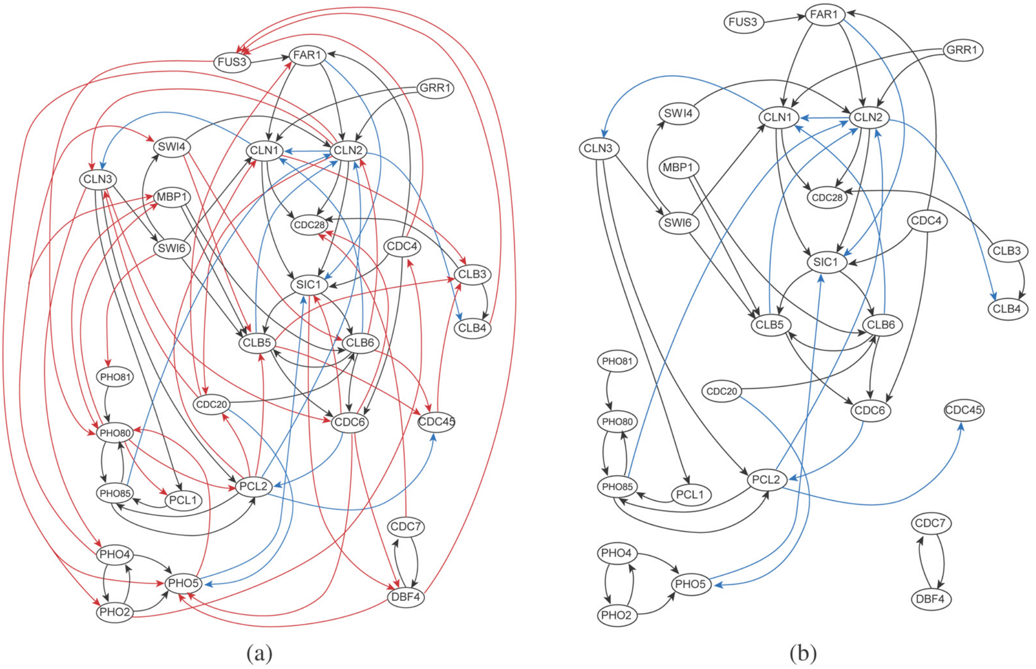


Fig. 4. Resulting networks based on the proposed method with a dynamic Bayesian network model. The black and red edges agree and do not agree with the KEGG information, respectively. The blue edges also do not agree with the KEGG information but are added as new information. (a) Resulting network with edges that appear in two or more estimated networks out of four networks. (b) An informative network obtained by this analysis. This network is obtained by removing the red edges from the network in (a).

Table 2
Updated relationships in the cell-cycle gene network based on the proposed method

Updated edge	Op.	Data	Updated edge	Op.	Data
SWI4 → *CDC20*	add	1	*CDC45* → *CDC6*	add	3
CLN1 → *CLN3*	add	1	*PCL1* → *CDC6*	add	3
CDC20 → *PHO5*	add	1	*CDC6* → *PCL2*	add	3
CLB6 → *CLN1*	add	1	*DBF4* → *CLB3*	add	3
PHO5 → *PHO81*	add	1	*PCL2* → *PCL1*	add	3
CLN2 → *CLN1*	add	1	*FUS3* → *PCL1*	add	4
CDC45 → *CDC20*	add	1	*CLB5* → *CLB4*	add	4
CLN2 → *CLB4*	add	2	*CDC6* → *PCL1*	add	4
FAR1 → *SIC1*	add	2,3	*PHO85* → *CLN1*	add	4
PHO5 → *SIC1*	add	2	*DBF4* → *CLB4*	add	4
PCL2 → *CLN2*	add	2,3	*DBF4* → *CLB5*	add	4
CLB5 → *CLN2*	add	2	*CDC45* → *CLB4*	add	4
PCL2 → *CDC45*	add	2,3	*CLN2* → *PCL1*	add	4
FAR1 → *DBF4*	add	3	*PHO85* → *CLN2*	add	4

The column “Op.” means which operation is done. The numbers in “Data” represent time series data we used to update (1: alpha, 2: cdc15, 3: cdc28, 4: elu).

black and blue edges, which can be considered as an informative part of the estimated network in Fig. 4(a).

The results of this analysis are summarized as follows:

- An edge “*CLB6* → *CLN1*”, which is added as new information, is missing information of KEGG. In fact, Li and Cai [26] indicated this relationship.
- In the cell-cycle network in KEGG, the edges “*CLN3* → *PCL1-PHO85*” and “*CLN3* → *PCL2-PHO85*” are shown as possible but unknown relations. In this analysis, we used these edges as known relations and could find them in the estimated networks. We may conclude that the gene expression data support these relationships.
- In the cell-cycle network in KEGG, the genes, *PHO2, 4, 5, 80, 81, 85, PCL1, 2* and the other genes are separated, except for the edges that connect between *CLN3* and *PCL* complexes mentioned above. However, from Fig. 4(a), we observed that there are a lot of interactions between *PHO*, *PCL* genes and the other genes.
- In Table 2, the relationships “*PHO85* → *CLN1*” and “*PHO85* → *CLN2*” are added as new information. Actually, *CLN1*, *CLN2* (cyclin, G1/S-specific) and *PHO85* (cyclin-dependent protein kinase) regulate cell-cycle progression and *PHO85* is required to start the cell-cycle in the absence of *CLN1* and *CLN2*, see e.g. [2]. Therefore, it is natural that the relationships among *CLN1*, *CLN2* and *PHO85* are observed from expression data.
- There are several added edges between nodes that are not directly connected. For example, “*CLN1* → *CLN3*”, “*CLB6* → *CLN1*”, “*PHO5* → *PHO81*”, “*FAR1* → *SIC1*” among others have relatively shorter paths than the edges in the KEGG network.
- The proposed method adds “*CDC45* → *CDC6*” to the database information in Table 2. In fact, the cell-cycle network in KEGG indicates that *CDC6* (cell division control protein), *CDC45* (related to chromosomal DNA replication), MCM complex and ORC complex form a protein complex, but *CDC6* and *CDC45* are not directly connected. Since we omitted MCM and ORC complexes in this analysis, we presume the proposed method added the relationship between *CDC6* and *CDC45* to the database. However, the added information “*CDC45* → *CDC6*” suggests that there may be some relationships between *CDC45* and *CDC6* that are partially correct.
- *CLB3* (cyclin-dependent protein kinase regulator activity) and *CLB4* (cyclin, G2/M-specific) do not connect to the other genes in the cell-cycle network in KEGG. On the other hand, by using the proposed method, we observed that some relationships among *CLB3*, *CLB4* and other genes, especially *CLN2* and *CLB5*, are estimated and added as new information. In fact, in the formation of mitotic spindles, it is known that *CLB5* works together with *CLB3* and *CLB4*, see e.g. [32].
- In this paper, we focus on the transcriptional regulations between genes. However, it is a possible case that the protein–protein interactions are estimated in our model. Although the protein–protein interactions are represented as undirected edges rather than directed edges, the estimated edges corresponding to protein–protein interactions could be considered as true positives. In the 92 estimated edges of Fig. 4(a), 23 edges are compiled as protein–protein interactions in the DIP database [31]. Also, an updated edge “*PHO85* → *CLN1*” in Table 2 is complied as a protein–protein interaction in DIP.
- The all update operations done in this analysis were “addition of the candidate relations as new information to the database”. Since the cell-cycle network in KEGG was established by collecting highly reliable information, our results are reasonable and the updated information in Fig. 4(b) and Table 2 could be considered as candidates for new findings.

4. Discussion

In this paper, we proposed a statistical model for simultaneously learning the database information and gene networks from gene expression data and a given biological knowledge database. The proposed method realizes a self-repairing system for biological knowledge databases that can revise database information by considering information on gene expression data. Three models, the Bayesian network model, the prior probability of the graph and the database information model, are unified as one statistical model based on Bayesian statistics. As shown in the Monte Carlo simulations and the real data example, the proposed method can recover the missing information and can improve the accuracy of the database.

We consider the following topics as our future work: (1) We regarded the database information as binary data. However, information contained in databases can have a varying degree of confidence. Therefore, the confidence level should be used to construct the prior probability of the graph. (2) In this paper, we focused on how to use database information for estimating a gene network. However, how to create database information by using various types of genomic data such as protein–protein interaction data, binding site information and so on, is a separate problem. (3) The transition probabilities of the database information model, denoted by p_h and p_m in Section 2.4, are actually parameters and are related to the reliability of the database information. (4) The reliability of the update information could be measured by the bootstrap method [9,10]. We could construct statistical criteria to choose them. We would like to discuss those problems in our future paper.

Acknowledgments

The authors would like to thank the reviewer and an associate editor, Prof. Grace Shieh, for their constructive comments and suggestions that improved the quality of the paper considerably.

References

[1] T. Akutsu, S. Miyano, S. Kuhara, Identification of genetic networks from a small number of gene expression patterns under the Boolean network model, Pac. Symp. Biocomput. 4 (1999) 17–28.

[2] B. Andrews, V. Measday, The cyclin family of budding yeast: abundant use of a good idea, Trends Genet. 14 (1998) 66–72.

[3] A. Bernard, A.J. Hartemink, Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data, Pac. Symp. Biocomput. 10 (2005) 459–470.

[4] T. Chen, H. He, G. Church, Modeling gene expression with differential equations, Pac. Symp. Biocomput. 4 (1999) 29–40.

[5] A.C. Davison, Approximate predictive likelihood, Biometrika 73 (1986) 323–332.

[6] C. De Boor, A Practical Guide to Splines, Springer, Berlin, 1978.

[7] M.J.L. De Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, S. Miyano, Inferring gene regulatory networks from time-ordered gene expression data of *Bacillus subtilis* using differential equations, Pac. Symp. Biocomput. 8 (2003) 17–28.

[8] M.J.L. De Hoon, Y. Makita, S. Imoto, K. Kobayashi, N. Ogasawara, K. Nakai, S. Miyano, Predicting gene regulation by sigma factors in Bacillus subtilis from genome-wide data, Bioinformatics 20 (2004) i101–i108.

[9] B. Efron, Bootstrap methods: Another look at the jackknife, Ann. Statist. 7 (1979) 1–26.

[10] B. Efron, E. Halloran, S. Holmes, Bootstrap confidence levels for phylogenetic trees, Proc. Natl. Acad. Sci. USA 93 (1996) 13429–13434.

[11] P.H.C. Eilers, B. Marx, Flexible smoothing with *B*-splines and penalties (with discussion), Statist. Sci. 11 (1996) 89–121.

[12] N. Friedman, M. Goldszmidt, Learning Bayesian networks with local structure, in: M.I. Jordan (Ed.), Graphical Models, Kluwer Academic Publishers, 1998, pp. 421–459.

[13] N. Friedman, M. Linial, I. Nachman, D. Pe’er, Using Bayesian network to analyze expression data, J. Comp. Biol. 7 (2000) 601–620.

[14] N. Friedman, K. Murphy, S. Russell, Learning the structure of dynamic probabilistic networks, in: Proc. 14th Conf. Uncertainty in Art. Intel., 1998, pp. 139–147.

[15] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N.J. Krogan, S. Chung, A. Emili, M. Snyder, J.F. Greenblatt, M. Gerstein, A Bayesian networks approach for predicting protein–protein interactions from genomic data, Science 302 (2003) 449–453.

[16] A.J. Hartemink, D.K. Gifford, T.S. Jaakkola, R.A. Young, Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks, Pac. Symp. Biocomput. 6 (2001) 422–433.

[17] A.J. Hartemink, D.K. Gifford, T.S. Jaakkola, R.A. Young, Combining location and expression data for principled discovery of genetic regulatory network models, Pac. Symp. Biocomput. 7 (2002) 437–449.

[18] S. Imoto, T. Goto, S. Miyano, Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression, Pac. Symp. Biocomput. 7 (2002) 175–186.

[19] S. Imoto, S. Kim, T. Goto, S. Aburatani, K. Tashiro, S. Kuhara, S. Miyano, Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network, J. Bioinform. Comp. Biol. 1 (2003) 231–252.

[20] S. Imoto, S. Konishi, Selection of smoothing parameters in B-spline nonparametric regression models using information criteria, Ann. Inst. Statist. Math. 55 (2003) 671–687.

[21] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, S. Miyano, Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks, J. Bioinform. Comp. Biol. 2 (2004) 77–98.

[22] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, M. Hattori, The KEGG resources for deciphering the genome, Nucleic Acids Res. 32 (2004) D277–D280.

[23] S. Kim, S. Imoto, S. Miyano, Inferring gene networks from time series microarray data using dynamic Bayesian networks, Brief. Bioinform. 4 (2003) 228–235.

[24] S. Kim, S. Imoto, S. Miyano, Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data, Biosystems 75 (2004) 57–65.

[25] S. Konishi, T. Ando, S. Imoto, Bayesian information criteria and smoothing parameter selection in radial basis function networks, Biometrika 91 (2004) 27–43.

[26] X. Li, M. Cai, Recovery of the yeast cell cycle from heat shock-induced G(1) arrest involves a positive regulation of G(1) cyclin expression by the S phase cyclin Clb5, J. Biol. Chem. 274 (1999) 24220–24231.

[27] S. Liang, S. Fuhrman, R. Somogyi, REVEAL, a general reverse engineering algorithm for inference of genetic network architectures, Pac. Symp. Biocomput. 3 (1998) 18–29.

[28] K. Murphy, S. Mian, Modelling gene expression data using dynamic Bayesian networks, Technical Report, Computer Science Division, University of California, Berkeley, CA, 1999.

[29] N. Nariai, S. Kim, S. Imoto, S. Miyano, Using protein–protein interactions for refining gene networks estimated from microarray data by Bayesian networks, Pac. Symp. Biocomput. 9 (2004) 336–347.

[30] D. Pe’er, A. Regev, G. Elidan, N. Friedman, Inferring subnetworks from perturbed expression profiles, Bioinformatics 17 (2001) S215–S224.

[31] L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, D. Eisenberg, The database of interacting proteins: 2004 update, Nucleic Acids Res. 32 (2004) D449–D451.

[32] E. Schwob, K. Nasmyth, CLB5 and CLB6, a new pair of B cyclins involved in DNA replication in *Saccharomyces cerevisiae*, Genes Dev. 7 (1993) 1160–1175.

[33] E. Segal, H. Wang, D. Koller, Discovering molecular pathways from protein interaction and gene expression data, Bioinformatics 19 (2003) i264–i272.

[34] E. Segal, R. Yelensky, D. Koller, Genome-wide discovery of transcriptional modules from DNA sequence and gene expression, Bioinformatics 19 (2003) i273–i282.

[35] I. Shmulevich, E.R. Dougherty, S. Kim, W. Zhang, Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks, Bioinformatics 18 (2002) 261–274.

[36] R. Somogyi, C.A. Sniegoski, Modeling the complexity of genetic networks: Understanding multigene and pleiotropic regulation, Complexity 1 (1996) 45–63.

[37] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, B. Futcher, Comprehensive identification of cell cycleregulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization, Mol. Biol. Cell 9 (1998) 3273–3297.

[38] Y. Tamada, S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara, S. Miyano, Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection, Bioinformatics 19 (2003) ii227–ii236.

[39] L. Tinerey, J.B. Kadane, Accurate approximations for posterior moments and marginal densities, J. Amer. Statist. Assoc. 81 (1986) 82–86.

[40] E.P. van Someren, L.F.A. Wessels, E. Backer, M.J.T. Reinders, Genetic network modeling, Pharmacogenomics 3 (2002) 507–525.