IADIS International Journal on Computer Science and Information Systems
Vol. 4, No. 1, pp. 54-63
ISSN: 1646-3692

Sensitivity and Specificity of Inferring Genetic Regulatory Interactions with the VBEM Algorithm

Isabel M. Tienda-Luna[†], **Maria C. Carrion Perez**[♣], **Diego P. Ruiz Padillo**[♣], **Yufang Yin**[‡], **Yufei Huang**[‡]

[†] Department of Electronics and Computer Science, University of Granada, Spain
[♣] Department of Applied Physics, University of Granada, Spain
[‡] Department of Electrical and Computer Engineering, University of Texas at San Antonio (UTSA), USA

Abstract

In this paper we perform a study of the performance of the VBEM algorithm proposed in [19]. The VBEM is a Bayesian approach for reconstructing gene regulatory networks (GRNs) based on microarray data. We focus on a variable selection formulation and develop a solution by a variational Bayes Expectation Maximization (VBEM) learning rule. The major advantage of the VBEM solution over Monte Carlo sampling based approach is its lower computational complexity. This makes it appealing for uncovering large networks. The suitability of the proposed algorithm to infer large networks is studied in terms of its *ROC* curves.

Keywords: Gene networks, Microarray data, Bayesian inference, Variational Bayesian Expectation Maximization, ROC curves

1 Introduction

Gene Regulatory Networks are collections of gene-gene regulatory relations in a genome and are models that display causal relationships between gene activities. With the completion of the Human Genome Project and successful sequencing genomes of many other organisms, emphasis of post-genomic research has been extended to the understanding of functions of genes [14]. We investigate in this paper reverse engineering Gene Regulatory Networks (GRNs) based on time series microarray data. GRNs are the functioning circuitry in living organisms at the gene level. They display the regulatory relationships among genes in a cellular system. These regulatory relationships involve directly and indirectly in controlling the production of protein and directing metabolic processes. Understanding GRNs can provide new ideas for treating complex diseases and breakthroughs for designing new drugs. Mathematically, reverse engineering GRNs is a traditional inverse problem, whose solutions require proper modelling and learning form data. Drawbacks like the big amount of unknowns variables, the small sample size or the inherent experimental defects call for powerful mathematic modelling together with reliable inference as long as the capability to integrate different types of relevant data.

In the literature, many different models have been proposed for both static and cell cycle networks and they include probabilistic Boolean networks [16], (dynamic) Bayesian networks [8, 9, 11, 15], differential equations [7], and others [1,17]. Unlike in the case of static experiments, extra attention is needed in modelling

*I. M. Tienda-Luna, D. P. Ruiz Padillo and M.C. Carrion Perez are supported by the Spanish MCyT under project TEC2007-68030-C02-02/TCM. Y. Huang is supported by an NSF Grant CCF-0546345.

the time series experiments to account for temporal dependency between samples. Such time series models can in turn complicate the inference, thus making the task of reverse engineering even tougher than it already is. In this paper, we apply dynamic Bayesian networks (DBNs) to model time series microarray data. DBNs have been applied to reverse engineering GRNs in the past [2, 3, 8, 9]. Differences among the existing work are the specific adopted models for gene regulations and the detailed inference objectives and algorithms. To learn the proposed DBNs from time series data, we aim at soft Bayesian solutions, i.e., the solutions that provide the a posterior probabilities (APPs) of the network topology. This requirement separates the proposed solutions with most of the existing approaches such as greedy search and simulated annealing based algorithms, all of which produce only point estimates of the networks and are considered as 'hard' solutions. The advantage of soft solutions has been demonstrated in digital communications [21]. In the context of GRNs, the APPs from the soft solutions provide valuable measurements of confidence of inference, which is difficult with hard solutions. Moreover, the obtained APPs can be used for Bayesian data integration. Soft solutions including Markov chain Monte Carlo (MCMC) sampling [11, 20] and variational Bayesian expectation maximization (VBEM) [2, 3] have been proposed for learning the GRNs. However, MCMC sampling is only feasible for small networks due to its high complexity. In contrast, VBEM has been shown to be much more efficient. However, the VBEM algorithm in [2,3] was developed only for parameter learning. It therefore cannot provide the desired APPs of topology. In [19] we proposed a new variational Bayesian EM (VBEM) algorithm that can learn both parameters and topology of a network. The algorithm still maintains the general feature of other VBEM algorithms for having low complexity, thus appropriate for learning large networks. In addition, it estimates the APPs of topology directly. In this paper we present a study of the performance of the proposed VBEM algorithm in terms of its sensitivity and specificity. The rest of the paper is organized as follows: In Section 2, the problem formulation is presented. In Section 3, objectives on learning the networks are discussed and the VBEM algorithm is developed. In section 4, the test results of the proposed VBEM on the simulated networks and yeast cell cycle data are provided.

2 Problem Formulation

In this paper, the objective is to uncover gene regulatory networks using microarray data so we start considering a microarray that measures the expression level of G genes evenly sampled at $N+1$ consecutive time instances. We define a random variable matrix $\mathbf{Y} \in \mathcal{R}^{G\times(N+1)}$ with the (i,n)th element $y_i(n-1)$ denoting the expression level of gene i measured at time $n1$. We further assume that the gene regulation yields to a linear regulatory relationship and follows a first-order time-homogeneous Markov process. This assumption may be insufficient but will facilitate the modeling and inference. Finally, we define a $G \times 1$ binary column vector $\mathbf{b}_i(n)$ of gene i, whose jth element is 1 if gene j regulates the target gene i and 0 else. The set of $\mathbf{b}_i(n), i, n$ is treated as latent variable which identifies the gene regulatory network structure. Based on the above definition and assumption, gene regulations for gene i can be expressed as follows

$$y_i(n) = \mathbf{w}_i \otimes \mathbf{y}(n-1) \otimes \mathbf{b}_i(n-1) + e_i(n), \quad n = 1, 2, \cdots, N \tag{1}$$

where $\otimes$ represents Kronecker product, $\mathbf{y}(n-1) = [y_1(n-1), \cdots, y_G(n-1)]^\top$, $\mathbf{w}_i \in \mathcal{R}^{G\times 1}$ is the weight vector independent of time n and $e_i(n)$ is white Gaussian noise with variance σ_i^2. The weight vector is very informative and provides the degree and the types of the regulation, e.g., up or down regulation (positive or negative). To be consistent with $\mathbf{w}_i$, $\mathbf{b}_i(n)$ is also time invariant and notated as $\mathbf{b}_i$. The noise variable is introduced to account for modeling and experimental errors.

An equivalent form of (1) can be further expressed in a more compact vector-matrix form as

$$\mathbf{y}_i = \mathbf{R}\mathbf{W}_i\mathbf{b}_i + \mathbf{e}_i \tag{2}$$

where $\mathbf{R} = \mathbf{T}\mathbf{Y}^\top$ with $\mathbf{T} = \begin{pmatrix} 1 & & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{pmatrix}$, $\mathbf{y}_i = [y_i(1), \cdots, y_i(N)]^\top$, $\mathbf{W}_i$ =diag$\{\mathbf{w}_i\}$, $\mathbf{b}_i = [b_i(1), \cdots, b_i(G)]^\top$

and $\mathbf{e}_i = [e_i(1), e_i(2), \cdots, e_i(N)]^\top$. Given a set of Microarray data $\mathbf{Y}$, our objective is to uncover gene regulatory networks, i.e., to determine $\mathbf{b}_i$ for all i. Note $\mathbf{W}_i$ and σ_i^2 are also unknown.

If we analyze expression 2 carefully, we can see that the proposed model has the same structure than the ones proposed for multiuser detection [21]. Therefore they are analogous problems so we can apply all our knowledge about multiuser detection to solve this new problem. Thus, in this paper we are going to analyze the performance on a Bayesian algorithm proposed in [19]. There are two main problems to deal with when we are looking for a solution. The first one is related to the fact that the weights and the noise variance are unknown. Thus, the proposed algorithm must be able to estimate them. On the other hand, the sample size N of the microarray data is usually much smaller than the number of genes G, this makes the problem we are dealing with ill-conditioned and the algorithm must be able to circumvent this drawback. This will be one of the issues to take into account when the performance of the algorithm is studied.

3 VBEM algorithm

If we are interested in applying a Bayesian solution to the problem presented in the previous section, we need to obtain the posterior distribution of variables $\mathbf{b}_i$. Nevertheless, it is not possible to calculate this distribution since it has associated the resolution of some integrals that are not analytically tractable. Thus, the calculation must be done numerically, using an approximation. The easiest way to approximate the integrals is to evaluate the integrand using an estimator for the parameters like the MAP (Maximum a posteriori). Another possibility is to estimate the integral numerically evaluating the integrand in different values (samples) using Monte Carlo Methods [10]. The higher the number of samples, the more exact the result provided by the method is. In spite of its good performance, the main drawback of this approach is its high computational complexity. Another option is to use the Laplace Method. An alternative way to approximate the integral is to use a Variational approach [4]. The key point of Variational Methods is the approximation of the integral by a simpler function that is mathematically treatable by the calculation of a lower or upper bound. In this way, the integration is reduced to a simpler problem: the optimization of a bound trying to make this bound as closer as possible to its true value.

The necessity of a method with low computational complexity leads us to propose a solution based on the VBEM algorithm [2] given the possibility to approximate those posterior distributions in a simple way by optimizing a lower bound of the marginal likelihood using an iterative process.

The optimal solution to the proposed problem can be written using a MAP criterion as:

$$\hat{b}_i(l) = argmax_{b_i(l)} p(b_i(l)|\mathbf{y}_i) \quad l = 1, \cdots, G \tag{3}$$

Where $p(b_i(l)|\mathbf{y}_i) \propto p(\mathbf{y}_i|b_i(l))p(b_i(l))$ and $p(\mathbf{y}_i|b_i(l))$ is the probability function obtained using the Bayesian rule:

$$p(\mathbf{y}_i|b_i(l)) = \sum_{(\mathbf{b}_i)_{-l}} p(\mathbf{y}_i|\mathbf{b}_i)p((\mathbf{b}_i)_{-l}) \tag{4}$$

where the summation is done over all possible configurations of vector $\mathbf{b}_i$ except for $b_i(l)$ and $(\mathbf{b}_i)_{-l}$ represents a vector of size $(G-1)\times 1$ that contains all the latent variables except for $b_i(l)$.

Apparently, the solution given by 4 has a complexity that grows exponentially with the number of genes, so this solution becomes computationally prohibitive when we are working with a big number of genes. Thus, if we want face these drawbacks; we need to be able to develop efficient algorithms able to diminish the computational complexity when working with big gene networks. In [20] an approach to approximate the Bayesian inference was proposed. This algorithm is a 'Reversible jump Markov chain Monte Carlo' (RJMCMC). Although their results were good, its good performance is limited to cases where G is not too big since its computational burden becomes very high as G increases (real situations).

Thus, in order to decrease the computational complexity related to the computation of 4, we approximate the distribution of variable $\mathbf{b}_i$ by a Gaussian. The main advantage of this approximation is that it permits an analytical treatment of the problem, reducing in this way the computational burden.

The marginal likelihood $p(\mathbf{y}_i|\mathbf{b}_i)$ is obtained via integration over the unknown parameters in the following way:

$$p(\mathbf{y}_i|\mathbf{b}_i) = \int\int p(\mathbf{y}_i \mid \mathbf{w}_i, \sigma_i^2, \mathbf{b}_i) p(\mathbf{w}_i, \sigma_i^2 \mid \mathbf{b}_i) d\sigma_i^2 d\mathbf{w}_i \tag{5}$$

where $p(\mathbf{w}_i, \sigma_i^2 \mid \mathbf{b}_i)$ is the prior of the parameters. We choose a *Gaussian-Inverse-Gamma* distribution as a prior [4]:

$$p(\mathbf{w}_i, \sigma_i^2 \mid \mathbf{b}_i) = \mathcal{N}_{\mathbf{w}_i}(0, \sigma\tilde{\mathbf{R}}) \mathcal{IG}_{\sigma_i}(\nu_0, \gamma_0) \tag{6}$$

where $\tilde{\mathbf{R}}^{-1} = \mathbf{H}^T\mathbf{H}$ with $\mathbf{H} = \mathbf{R}\mathbf{B}_i$, $\mathbf{B}_i = diag\mathbf{b}_i$ and ν_0 and γ_0 are small, real and positive values. With the latter configuration, the parameter and latent variables distributions are conjugate of the approximate posterior distributions.

Given the prior information and the observations, the VBEM algorithm calculates the lower bound of the marginal likelihood using the Jensen inequality and the mean field approximation [6]:

$$\begin{aligned} \ln p(\mathbf{y}_i) &= \ln \int d\boldsymbol{\theta}_i d\mathbf{b}_i p(\mathbf{b}_i, \mathbf{y}_i, \boldsymbol{\theta}_i) \\ &\geq \int d\boldsymbol{\theta}_i d\mathbf{b}_i q(\boldsymbol{\theta}_i, \mathbf{b}_i) \ln \frac{p(\mathbf{b}_i, \mathbf{y}_i, \boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i, \mathbf{b}_i)} \end{aligned} \tag{7}$$

where $\boldsymbol{\theta}_i = [\mathbf{w}_i^\top \ \sigma_i^2]^\top$ and $q(\boldsymbol{\theta}_i)$ and $q(\mathbf{b}_i)$ are the approximated distribution to determine. The VBEM algorithm calculates those approximated distributions by maximizing iteratively 7 with respect to $q(\boldsymbol{\theta}_i)$ and $q(\mathbf{b}_i)$ accordingly to the variational Bayesian learning rule [2]. This learning rule has two parts that are explained in the following subsections.

3.1 VBE step

The posterior distribution $p(\mathbf{b}_i, \boldsymbol{\theta}_i|\mathbf{y}_i)$ has second order interaction terms. As a result, the integration in the latent variables ($\mathbf{b}_i$) and parameters is not analytically possible. However, since the model we are using is Conjugate-Exponential, we can overcome this difficulty developing a general VBEM algorithm based on the properties of the Conjugate-Exponential models [2]. Thus, the first step is to specify the prior distribution of the parameters and latent variables. To do so, we start identifying the prior conjugate distributions and rewriting them according to the Conjugate-Exponential model. This task is accomplished taking into account that these prior distributions must satisfy the two main conditions of the Conjugate-Exponential models.

Once the prior distributions are known and properly written, the VBE step is performed in order to find the posterior variational distribution of the latent variables. To do so, we apply the variational Bayesian theory for Conjugate Exponential models using as a starting point an arbitrary distribution on the parameters:

$$q(\mathbf{b}_i) = \mathcal{N}(\mathbf{b}_i|\boldsymbol{\mu}_{\mathbf{b}_i}, \boldsymbol{\Sigma}_{\mathbf{b}_i}) \tag{8}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{b}_i} &= \boldsymbol{\Sigma}_{\mathbf{b}_i}(\sigma_0^{-2}\boldsymbol{\mu}_0 + \mathbf{f}) \\ \boldsymbol{\Sigma}_{\mathbf{b}_i} &= (\sigma_0^{-2}\mathbf{I}_G + \mathbf{D})^{-1} \end{aligned} \tag{9}$$

con

$$\begin{aligned} \mathbf{D} &= \mathbf{B} \otimes \left[(\mathbf{m}_{\mathbf{w}_i}\mathbf{m}_{\mathbf{w}_i})^\top \langle\sigma_i^{-2}\rangle_{q(\boldsymbol{\theta}_i)} + \mathbf{A}^{-1}\right] \\ \mathbf{f}^\top &= \mathbf{y}_i^\top \mathbf{R} diag(\mathbf{m}_{\mathbf{w}_i}) \langle\sigma_i^{-2}\rangle_{q(\boldsymbol{\theta}_i)} \\ \mathbf{B} &= \mathbf{R}^\top\mathbf{R} \\ \mathbf{A} &= \mathbf{I}_G + \mathbf{K} \\ \mathbf{K} &= \mathbf{B} \otimes (\boldsymbol{\Sigma}_{\mathbf{b}_i} + \boldsymbol{\mu}_{\mathbf{b}_i}\boldsymbol{\mu}_{\mathbf{b}_i}^\top) \end{aligned} \tag{10}$$

The marginal posterior distribution $p(b_i(l)|\mathbf{y}_i)$ is updated in next way till the algorithm converges:

$$p(b_i(l)|\mathbf{y}_i) = \mathcal{N}(m_{\mathbf{b}_i}(l), \Sigma_{\mathbf{b}_i}(l,l)) \tag{11}$$

and finally the decision on $\mathbf{b}_i(l)$ is done in the following way:

$$\hat{b}_i(l) = \mathrm{a}rgmax_{b_i(l)} p(b_i(l)|\mathbf{y}_i) \tag{12}$$

3.2 VBM step

In the VBM step, the objective is to calculate the distributions of the unknown parameters taking into account the distributions calculated in the previous VEB step for the latent variables. In the first place we calculate $q(\boldsymbol{\theta}_i)$ in the following way:

$$q(\boldsymbol{\theta}_i) = \mathcal{N}(\mathbf{w}_i|\mathbf{m}_{\mathbf{w}_i}, \boldsymbol{\Sigma}_{\mathbf{w}_i})\mathcal{IG}\left(\frac{\alpha}{2}, \frac{\beta}{2}\right) \tag{13}$$

where

$$\begin{aligned} \mathbf{m}_{\mathbf{w}_i} &= (\mathbf{I}_G + \mathbf{K})^{-1}(\mathbf{y}_i^\top \mathbf{R}\mathbf{M}_x)^\top \\ \boldsymbol{\Sigma}_{\mathbf{w}_i} &= \sigma_i^2(\mathbf{I}_G + \mathbf{K})^{-1} \\ \frac{\alpha}{2} &= \frac{N(\eta+1)-2}{2} \\ \frac{\beta}{2} &= -\frac{c}{2} \end{aligned} \tag{14}$$

with

$$\begin{aligned} \mathbf{M}_x &= diag(\mathbf{m}_{\mathbf{b}_i}) \\ c &= \mathbf{y}_i^\top \mathbf{R}\mathbf{M}_x\mathbf{m}_{\mathbf{w}_i} - \mathbf{y}_i^\top \mathbf{y}_i - \nu_0 \end{aligned} \tag{15}$$

3.3 Lower Bound calculation ($\mathcal{F}$)

Finally, once the $q(\boldsymbol{\theta}_i)$ and $q(\mathbf{b}_i)$ have been updated, the lower bound of the distribution $\ln p(\mathbf{y}_i \mid \mathbf{b}_i)$ is computed and updated. This process is repeated until convergence. The lower bound is calculated using the Kulback-Leiber distance. The Kulback-Leiber measuresthe difference between two probability distributions. This distance is also known as relative entropy and mathematically can be expressed as:

$$\begin{aligned} \mathcal{F} &= \int d\boldsymbol{\theta}_i q(\boldsymbol{\theta}_i) \left[\int d\mathbf{b}_i q(\mathbf{b}_i) \ln \frac{p(\mathbf{b}_i, \mathbf{y}_i|\boldsymbol{\theta}_i)}{q(\mathbf{b}_i)} + \ln \frac{p(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)} \right] \\ &= -\int d\boldsymbol{\theta}_i q(\boldsymbol{\theta}_i) \mathrm{KL}\left[q(\mathbf{b}_i) \| p(\mathbf{b}_i, \mathbf{y}_i|\boldsymbol{\theta}_i)\right] - \mathrm{KL}\left[q(\boldsymbol{\theta}_i) \| p(\boldsymbol{\theta}_i)\right] \end{aligned} \tag{16}$$

4 Simulation results

4.1 Study of the performance through ROC

In this section, we present a study of the performance of the proposed VBEM algorithm in terms of the *ROC* (Receiver Operating Characteristics) curves. In general, a *ROC* curve is a graphical plot of the sensitivity versus (1 - specificity) for a binary classifier system as its discrimination threshold is varied. In our case the discrimination threshold is the APP. The sensitivity is a statistical measure of how well a binary classification test correctly identifies a condition and it can be computed as $TP/(TP + FN)$. On the other hand, the specificity is a statistical measure of how well a binary classification test correctly classifies cases

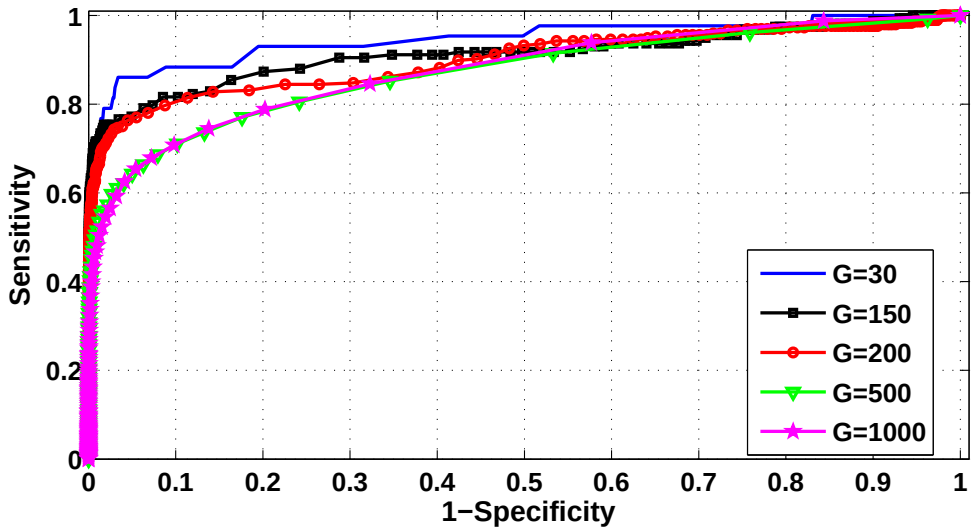

Figure 1: ROC curve for the VBEM algorithm using synthetic data with different settings

not belonging to that class and it can be calculated as $TN/(TN + FP)$. Both, sensitivity and specificity varies between 0 and 1. The best possible prediction method would yield a graph that was a point in the upper left corner of the ROC space, i.e. 100% sensitivity (all true positives are found) and 100% specificity (no false positives are found). A completely random predictor would give a straight line at an angle of 45 degrees from the horizontal, from bottom left to top right (the so-called 'line of no-discrimination'). To pursue our objective, we have created simulated networks of 30, 150, 200, 500 and 1000 genes. For each tested networks, we have collected only 30 time samples for each gene, which mimics the realistic small sample scenario. Regarding to the regulation process, each gene can have either none, one, two or three parents. Besides, the number of parents has been selected randomly for each gene. The weights associated to each regulation process have also been chosen randomly from an interval that contains the typical values estimated when working with microarray data. As for the nature of regulation (the sign of the weight), it has also been selected randomly. Finally the observations have been calculated using the linear Gaussian model proposed in 1. These observations have been taken once the system has reached stationarity and their values are also in the range of the observations corresponding to real microarray data. In Figure 4.1 we have plotted the ROC curves for the different settings. In order to construct these curves, we started by setting a threshold for the APPs. This threshold ρ is between 0 to 1 and it was used as in [1]: for each possible regulation relationship between two genes, if its APP is greater than ρ, then the link is considered to exist, whereas if the APP is lower than ρ, the link is not considered. We calculated the precision and the recall for each selected threshold between 0 and 1. We plotted the results in blue for the case with $G = 30$, black for $G = 150$, red for $G = 200$, green for $G = 500$ and magenta for $G = 1000$. As expected, the performance got worse as the number of genes increases. However, we can state that, in spite of the degradation, the performance of the algorithm is very good for all the settings even for the one where $G = 1000$ and $N = 30$. One measure of aforementioned degradation is shown in Table 1 where we calculated the area under each curve (AUC). As can be noticed, the area decreases as G increases but still is good enough even if the number of genes is 1000 and the number of samples only 30.

To study the scalability of the proposed algorithm, we have also calculated the ROC curves for simulated networks characterized by the following settings: $G_1 = 1000$ and $N_1 = 400$ (solid blue line), $G_2 = 500$ and $N_2 = 200$ (black squares), $G_3 = 250$ and $N_3 = 100$ (red stars) and $G_4 = 125$ and $N_4 = 50$ (green triangles). As can be noticed, the ratio G_i/N_i has been kept constant in order to maintain the proportion between the amount of nodes in the network and the amount of information (samples). The results represented in Figure

G	30	150	200	500	1000
AUC	0.9434	0.9097	0.8979	0.8498	0.8496

Table 1: Area under each curve in Figure 4.1, the higher the area, the better the performance of the algorithm

Setting	Errors Gibbs Sampling (500 samples)	Errors VBEM
G=5,N=5	126	26
G=5,N=10	5	1
G=20,N=10	N/A	6

Table 2: Comparison between the performance of the VBEM and Gibbs Sampling algorithms

4.1 shows a good enough scalability of the algorithm.

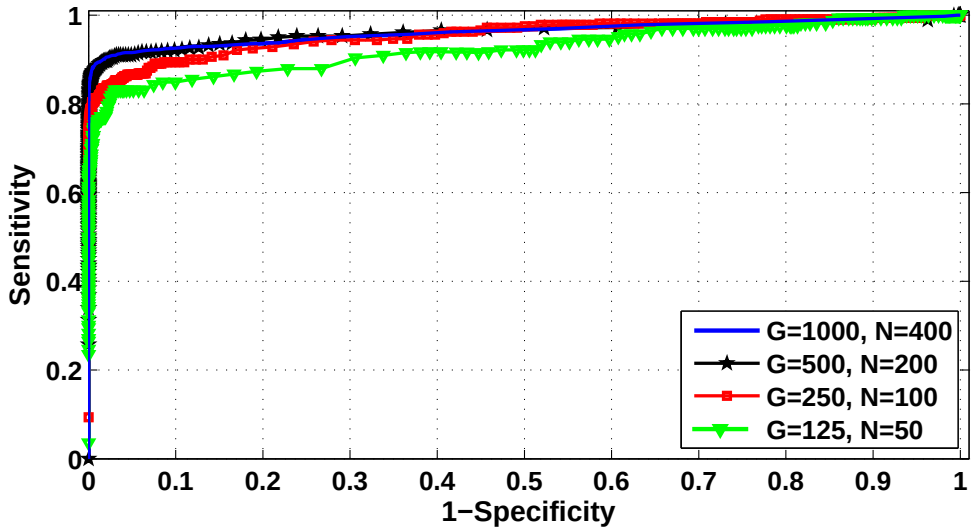

Figure 2: ROC curve for different setting maintaining the ratio G_i/N_i constant in order to show the scalability of the VBEM algorithm

4.2 Comparison with the Gibbs sampling

We also have done a comparison between the VBEM and the Gibbs sampling algorithm. To do such a comparison we have used synthetic data generated using a simulated network. We focused on a particular gene in the simulated networks. The gene was assumed to have two parents. In Table 2, we present the number of errors in 100 Monte Carlo tests. For the Gibbs sampling, 500 Monte Carlo samples are used. We tested the algorithms under different setting. From these results we can see that the proposed algorithm achieve better results with less computational complexity.

4.3 Test on real data

Finally, we show the results of applying the VBEM algorithm to real microarray data, specifically to the set reported in [18] for the cell cycle of the yeast *Saccharomyces cerevisiae*. This set contains information

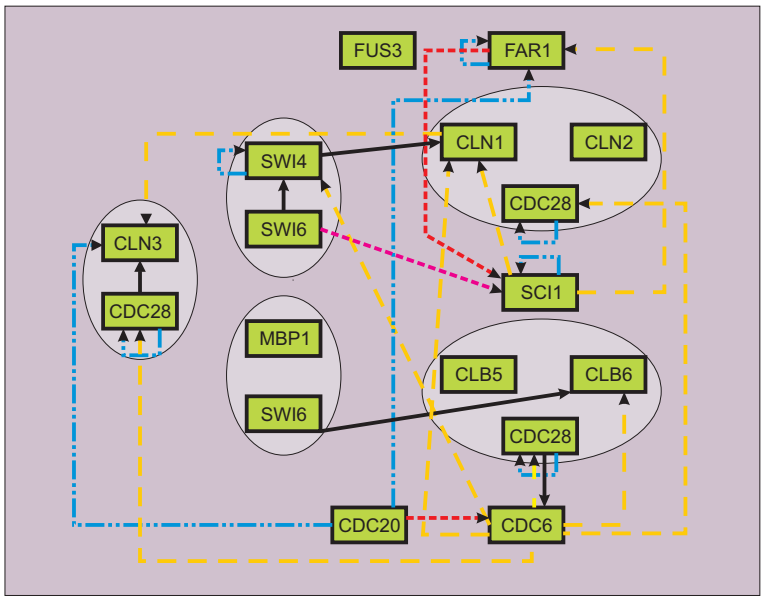

Figure 3: Predicted network using the VBEM. The solid lines represent direct relationships and the dotted lines represents indirect relationships (meaning that there are intermediate elements) in agreement with the KEGG. The dashed lines represent relationships predicted in the the opposite direction. The dashed-dotted lines represent relationships (mainly autorregulation ones) not confirmed by the KEGG.

about 62 genes and 17 samples. In order to synchronize the set of data, a CDC15 mutant sensitive to the temperature has been used. The cellular cycle under study lasts 85 minutes. For each gene, the data is represented by $\log_2\{(\text{expression in } t)/(\text{expression in the mixture set of control cells})\}$. It is worth noting that there are missing values in the data set and this fact indicates that the used signal is not strong enough. This must be taken into account when analyzing the results given by the algorithm and may justify until some point some discrepancies. The missing values are calculated using spline interpolation.

In Figure 3 we have plotted the estimated network for the G1 state of the cell cycle. We have compared the results obtained with the VBEM algorithm with the ones shown in the KEGG pathway [12]. We would like to point out here that the KEGG has been created using information from different sources and this fact must be kept in mind when analyzing the results obtained in this paper with the ones of the KEGG pathway. The only data source used to test the VBEM algorithm is microarray data and, thus, the expected results are somehow limited since the information used is also limited. We have used solid lines to represent the direct relationships confirmed by the KEGG pathway. The dotted lines are used for indirect relationships (meaning that there are intermediate elements) also confirmed by the KEGG. One example of this is the link between from SWI4 and CLN1. This relationship has a APP of 0.6757 and a weight of 0.0633 positive that indicates an upregulation as confirmed by the KEGG pathway. On the other hand, the dashed lines represent relationships confirmed by the KEGG but estimated by the VBEM algorithm in the wrong way (the sign of the estimated weight is opposite to the one predicted by the KEGG). An example of this situation is the link between the CLN1 and CLN3. Its APP is 0.7723 but the weight is negative and it should be positive according to the KEGG pathway. Finally, we have used dashed-dotted lines for those relationships (mainly autorregulation ones) not confirmed by the KEGG. Since the proposed algorithm intends to be the first step of a more complete approach, the main objective of this paper is to study its ability to uncover or identify regulation relationships and not direct associations in the KEGG. This is the reason why the dashed and dotted lines are also considered as successes of the algorithm. Thus, we can state that most of the associations reflected in the KEGG (Kyoto Encyclopedia of Genes and Genome) are predicted by the proposed algorithm.

5 Conclusions

In this paper we have studied the performance of the VBEM algorithm through simulations with synthetic date and the corresponding analysis of the ROC curves since the problem we have dealt with a classification one. In the first place we have demonstrated that the suitability of the proposed algorithm for large network. We have shown that, although its performance degrades when the ratio G_i/N_i increases, it remains appropriate for the real situations we normally deal with. On the other hand, the ROC curves have also been used to study the scalability of the algorithm since this is a problem worth considering in this field. Again, we have shown that the behavior of the proposed algorithm is suitable in spite of the slight degradation observed as the number of time samples is reduced. Finally, we have also shown using synthetic data the superiority of the VBEM algorithm in comparison to the Gibbs sampling algorithm. We have shown a reduction in the computational burden as well as a improvement of the results in terms of the number of errors. On the other hand, we have also shown the performance of the algorithm when it is used with real data. Specifically, we have used data from the cell cycle of the yeast *Saccharomyces cerevisiae* and we have compared the results with the ones shown in the KEGG pathway. We have found many agreements. Moreover, it is importante to notice the fact that the proposed algorithm is a soft Bayesian solution and thus, it provides with APP that can be used in processes of data integration from different sources providing in this way improved results [5].

References

[1] Bar-Joseph Z. (2004). Analyzing time series gene expression data *Bioinformatics*, vol. 20, no. 16, pp. 24932503

[2] Beal, M. J.(2003). *Variational algorithms for approximate Bayesian inference*. PhD thesis, The Gatsby Computational Neuroscience Unit, University College London.

[3] Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C., & Wild, D. L. (2004). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 20:1361–1372.

[4] Bernardo J. M. & Smith A. F.(1994). Bayesian Theory *John Wiley and Son Ltd,*

[5] Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D., & Davis, R. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, (2):65–73.

[6] Cover, T. M., & Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley.

[7] H. de Jong.(2002). Modeling and simulation of genetic regulatory systems: A literature review *Journal of Computational Biology*, vol. 9, no. 1 pp. 67103

[8] Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805.

[9] Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620.

[10] George, E. I. & McCulloch, R. (1993) Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88:881–889.

[11] Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–2282.

[12] Kegg: Kyoto Encyclopedia of Genes and Genome *http://www.genome.jp/kegg*

[13] Kim, S. Y., Imoto, S., & Miyano, S. (2003). Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics*, 4(3):228–235.

[14] Kitano, H. (2002). Looking beyond the details: a rise in system-oriented approaches in genetics and molecular biology. *Current Genetics*, 41:1–10.

[15] Pournara, I., & Wernisch, L. (2004). Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics*, 20(17):2934–2942.

[16] Shmulevich I., Dougherty E. R., Kim S., & Zhang W.(2002). Probabilistic Boolean networks:a rule-based uncertainty model for gene regulatory networks *Bioinformatics*, vol. 18, no. 1

[17] Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., & Friedman, N. (2003). Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166-76.

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

[19] Tienda-Luna, I. M.,Yin Y., Huang Y., Ruiz Padillo D.P., & Carrion Perez M. C. (2007). Uncovering Gene Networks Using Variational Bayesian Variable Selection. *Artificial Life. MIT Press, Special Issue on Systems Biology*, Accepted.

[20] Wang, J., Huang, Y., Wang, Y., & Zhang, J. (2006). Reverse engineering yeast gene regulatory networks using graphical models. In *Proceedings of IEEE International Conference on Acoustics,Speech, and Signal Processing*.

[21] Wang X. & Poor H. V. (2004). Wireless Communication Systems: Advanced Techniques for Signal Reception *Prentice Hall PTR*