MCMC Based Bayesian Inference for Modeling Gene Networks

Ramesh Ram and Madhu Chetty

Gippsland School of IT, Monash University,
Churchill, Victoria 3842, Australia

{Ramesh.ram, Madhu.chetty}@infotech.monash.edu.au

Abstract. In this paper, we apply Bayesian networks (BN) to infer gene regulatory network (GRN) model from gene expression data. This inference process, consisting of structure search and conditional probability estimation, is challenging due to the size and quality of the data that is currently available. Our previous studies for GRN reconstruction involving evolutionary search algorithm obtained a most plausible graph structure referred as Independence-map (or simply I-map). However, the limitations of the data (large number of genes and less samples) can result in many plausible structures that equally satisfy the data set. In the present study, given the network structures, we estimate the conditional probability distribution of each variable (gene) from the data set to deduce a unique minimal I-map. This is achieved by using Markov Chain Monte Carlo (MCMC) method whereby the search space is iteratively reduced resulting in the required convergence within a reasonable computation time. We present empirical results on both, the synthetic and real-life data sets and also compare our approach with the plain MCMC sampling approach. The inferred minimal I-map on the real-life yeast data set is also presented.

Keywords: Bayesian network, gene expression, MCMC, parameter estimation.

1 Introduction

Cellular processes are controlled by gene-regulatory networks (GRN). The invention of DNA microarrays, which measure the abundance of thousands of mRNA targets simultaneously, has made way for several computational methods that are currently used to learn the structure of gene-regulatory networks. Some of the computational methods such as Boolean, multiple regression and Bayesian methods have been extensively explored and reviewed [1]. Bayesian networks [2-4] were first applied to the problem of inferring genetic networks from microarray expression data [5-7]. Bayesian networks are interpretable and flexible models for representing probabilistic relationships between multiple interacting genes. At a qualitative level, the structure of a Bayesian network describes the relationships between these genes in the form of conditional independence relations. At a quantitative level, relationships between the interacting genes are described by conditional probability distributions. The probabilistic nature of this approach is capable of handling both biological and technical noise

and makes the inference scheme robust and allows the confidence in the inferred network structures to be estimated objectively. However, the application of BN learning to gene expression data in understanding the mechanism of GRN is particularly hard because the data sets are very sparse, typically containing only a few dozen samples but thousands of genes. In this case, our goal is to devise computational methods that consistently identify causal and dependence relationships between expressions of different genes.

The most common approach to learning of GRN based on BN consists of two separate problems: structure learning and parameter estimation. The structure of the network (gene-gene interaction) is unknown and the gene expression data is also incomplete. In such a case, one has to resort to structure search and the approximation of parameters. As the number of BN structures is astronomically large and the problem is NP-hard [8], these strategies for structure search have to be advanced and sophisticated. Further, since the sample size is small compared to the number of genes, there are many sub optimal models that can fit the data equally well [9]. The essential edges which are present in majority of these optimal structures are of significance and great importance from the point of view of GRN modeling. Having detected such edges (interactions) between pairs of genes from the inferred structures, important information from biological literature is used to substantiate the findings.

After the inferred structure is validated for biological plausibility, estimation of parameters (i.e. conditional probability distributions (CPD)) of the given GRN is carried out. Estimating CPDs involve specifying $P(X \mid pa(X))$ for each of the gene (variable) X where the term $pa(X)$ refers to parents of variable X in the given structure. Assuming that the inferred structure G , is an Independence-map (I-map) of a probability distribution P , we note that $I(G) \subseteq I(P)$ where $I(G)$ represents independence assertions in graph G and $I(P)$ is the independence assertions in the probability distribution P . Since G is an I-map of P , P factorizes according to joint probability distribution (JPD) given by equation (1).

$$P(X_1, \dots, X_n) = \prod P(X_i \mid pa(X_i)) \quad (1)$$

The network is a pair (G, P) where G is specified in edges and P is specified in CPDs. With several optimal graphs G equally representing the distribution P , $I(G)$ becomes a subset of $I(P)$ as shown in the Fig below implying that we can obtain $P(X_1, \dots, X_n)$ from G . Once we obtain P , it is possible to deduce a unique minimal I-map G . Removing any edge from the minimal G then induces conditional independencies that do not hold in P .

Unlike other methods [10, 11], we are modelling the gene expression to be continuous rather than discrete. Further, due to high dimensional data, exact computation of the CPDs is infeasible and computationally expensive. Hence, the joint distribution can

Fig. 1. Independence Assertions

be approximated by stochastic simulation commonly referred as 'sampling'. Using Monte Carlo algorithms based on random sampling, we can fit a distribution to the data and retain the samples. However, random sampling from the GRN may not be the best strategy since the state space is enormous with large number of samples needed to approximate the probabilities reasonably well. One way for picking the most representative samples and increase efficiency is to create a 'Markov Chain' in which each sample is selected using the previous sample resulting in the well known Monte Carlo Markov Chain (MCMC) methods and its variants [12-15].

In this paper, we propose a new approach to approximate the conditional probability distributions of complex GRN models with the use of a MCMC method. The proposed approach is essentially based on two novel ideas. The first is an efficient computation of CPDs based on the ordered ranking of Markov Blankets (MB). We choose MB for ranking, because our earlier work using MB scoring metric to search for a structure produced promising results. The genes with high scoring MBs tend to be more accurate allowing much faster convergence compared with a stationary distribution of the Markov chain. The second novelty is progressively reducing the space by clamping those variables whose samples have converged to a fixed distribution thereby allowing convergence over a narrower region. Empirical results are presented to illustrate the superiority of the approach over direct MCMC and random sampling. Studies are performed using not only the synthetic data sets (which allow variation of parameters) but also the real life *Saccharomyces cerevisiae* (yeast) [16] microarray dataset. The rest of the paper is structured as follows. In Section 2, a brief overview of Bayesian learning and MCMC sampling is given. Section 3 elaborates on the system and methods of the proposed method. Section 4 provides experiments and results. Finally, section 5 has concluding remarks on the paper and some future work.

2 Background

In this section, we briefly elaborate on the probability distribution and sampling for GRN with a focus on Gibbs sampling which is a type of Markov Chain Monte Carlo (MCMC) sampling.

2.1 Probability Distribution and Sampling

a) Probability distribution: A GRN based on Bayesian network specifies a probability distribution through a directed acyclic graph (structure) and a collection of conditional probability distribution (parameters) for each gene X_i in the graph G . The graph G captures conditional independence relationships in its edges. A gene (node) is conditionally independent of all other genes (nodes) in network given its Markov Blanket (parents, children, and children's parents). The probabilities summarize a potentially infinite set of circumstances that are not explicit in the model but rather appear implicitly in the probability. If each gene (variable) is influenced by at most k others and we have n random genes (variables), then we only need to specify $n \cdot 2^k$ probabilities instead of 2^n . Succinctly, conditional probability distribution shows the probability distributions over all values of gene X given the values of its parent genes. Conditional probability distribution of $X=x$ given $Y=y$ is given by equation 2.

$$P(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)} \quad (2)$$

If genes x and y are independent, then $P(x|y) = p(x)$ since $p(x,y) = p(x)p(y)$. The equation 2 above is repeated to condition X on all parent genes of X . The parents of gene X_i are all those genes that directly influence gene X_i from the set of genes $X_1, \dots, X_{i-1}$. Since large GRN models will have more parameters, the exact computation is, therefore intractable and in such cases simulation (sampling) technique becomes suitable for approximating conditional distribution. The structure G , necessary for sampling, is obtained by applying a structure search over the entire space of all possible structures. Hence, given structure G with genes $X = \{X_1, X_2, \dots, X_n\}$, we can draw a sample from the joint probability distribution as follows:

- (i) Instantiate randomly all except one of the genes, X_i
- (ii) Compute the probability distribution over the states of X_i , i.e. $P(X_i|X_1 \dots X_{i-1}, X_{i+1}, \dots, X_n)$
- (iii) From the probability distribution, randomly select a state of X_i

If all genes in the network except the gene X_i are instantiated, then due to the factorization of the joint probability distribution, the full conditional for a given gene in the DAG involves only a subset of genes participating in its Markov blanket (i.e. the set of parents, children and other parents of the children for the gene).

$$P(X_i|X_1 \dots X_{i-1}, X_{i+1}, \dots, X_n) = P(X_i|MB(X_i)) \quad (3)$$

Here, $MB(X_i)$ is the Markov Blanket of gene X_i . Since gene X_i is independent of rest of the genes in the network (except its Markov blanket), it is necessary to consider only the partial conditional conditioning on the Markov blanket.

Furthermore,

$$P(X_i|MB(X_i)) = P(X_i|Pa(X_i)) \prod P(Y_i|Pa(Y_i)) \quad (4)$$

Here, $Y_i, i = 1, \dots, k$ are the children of gene X_i .

b) Sampling: The sampling using Monte Carlo methods involves drawing n samples from the GRN with the instantiated genes fixed at their values as explained above. From these samples, the probability distributions are estimated based on frequency of occurrence of genes. Since our model involves continuous expression values, we plot these samples as a histogram and then *smooth* the histogram to give the probability density function of the genes. The instantiation of the genes is done using the distribution available from the data set. However, due to typically large number of genes in a GRN, random sampling methods are not suitable because they can be slow and the posterior distribution estimated cannot be reliable. Markov Chain Monte Carlo (MCMC) approach is suitable in such cases for approximating the difficult high dimensional distributions. From amongst the many MCMC methods available, we chose Gibbs sampler which results in obtaining samples asymptotically from the posterior distribution and can provide convergence in reasonable computation time. The Gibbs sampler is discussed in detail in the next section.

2.2 Markov Chains and Gibbs Sampling

A MCMC method such as Gibbs sampler which is applied for sampling probability distributions is based on constructing a Markov chain. Next, we briefly present the concept of Markov chain followed by the Gibbs sampling technique.

a) Markov Chain: The Markov chain includes the probability of transitioning the variables from their *current* state s to the *next* state s' based on the transition probability $q(s \rightarrow s')$. If the state distribution $\pi_t(s)$ describes the probability of genes being in state s at the t -th step of the Markov chain, then the stationary (equilibrium, invariant) distribution $\pi^*(s)$ will occur when $\pi_t = \pi_{t+1}$, i.e.

$$\forall s' \quad \pi(s') = \sum_s \pi(s) q(s \rightarrow s') \quad (5)$$

We note that the stationary distribution also satisfies the detailed balance equation (6) given below.

$$\forall s, s' \quad \pi(s) q(s \rightarrow s') = \pi(s') q(s' \rightarrow s) \quad (6)$$

No matter what the initial state distribution is, a Markov chain converges to $\pi^*(s)$ if it fulfills the following conditions: unique, aperiodicity and irreducibility. The condition of aperiodicity ensures that the chain can not get trapped in cycles or the state transition graph is connected. The irreducibility condition ensures that for any state, there is positive probability to visit all other states. An aperiodic and irreducible Markov chain is called ergodic [17] and ensures that every state must be reachable from every other and there can be no strictly periodic cycles.

Using Gibbs sampling, we aim to design a Markov chain whose stationary distribution is the target (desired) distribution such that gene X_i quickly converges to the stationary distribution irrespective of the initial distribution. From this, we then run the chain to produce a sample; throwing away the initial (burn-in) samples as these are likely to be influenced by the initial distribution. The sampling method for the target distribution π^* on χ_v constructs a Markov chain $S^0, S^1, \dots, S^k, \dots$ with $\pi^*(s)$ as equilibrium distribution. Since the distribution $\pi^*(s)$ is a unique equilibrium, and the Markov chain is ergodic, we have

$$\forall s \quad \pi^*(s) = \lim_{n \rightarrow \infty} \pi_n^*(s) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=m+1}^{m+n} \chi_v(S^i) \quad (7)$$

Where n is the number of iterations. The state of the chain obtained after a large number of steps is then used as a sample and its quality improves with the increase in the number of iterations. When a dynamic equilibrium is reached, the long-term fraction of time spent in each state is exactly its posterior probability for the given conditions. As number of iteration tends towards infinity, all statistically important regions of state space will be visited.

b) Gibbs sampling: To perform a MCMC simulation on GRN where the target distribution is the joint probability distribution, we design a Markov chain where each state is a full joint instantiation of the distribution (i.e. values are assigned to all variables). Hence, a transition is a move from one joint instantiation to another. The target sampling distribution $\pi^*(x)$ of the GRN is the posterior joint distribution $P(x | e)$ where x is the set of unknown variables and e is the set of evidence variables. It is typically the unknown we want to evaluate. Although sampling methods such as logic sampling [18], rejection sampling [19] and importance sampling [20] are available to sample $P(x|e)$, in the absence of evidence e or with the probability of evidence being small (i.e. if $P(e) \approx 0$), these algorithms result in many wasted samples. The Gibbs sampling overcomes these limitations as it specifically uses conditional distribution $P(s' | s)$ to define state transition rules.

In Fig 2, an example of Markov Chain for a 4 gene GRN is shown. We have specifically fixed the Gene B and D values and Gene A and C are varied to produce 4 states.

Fig. 2. Example Markov Chain for a toy 4-gene network. The genes B and D are instantiated as true while the genes A and C are false.

The working of the Gibbs sampling algorithm is shown by the flow chart in Fig. 3. Consider a GRN with n unknown variables $X_1, X_2, \dots, X_n$ which appears as input to the algorithm. We now recall that a gene X_i is independent of rest of the network given the variables in the Markov blanket (MB) of X_i , i.e.

$$P(X_i | X_2, \dots, X_n) = P(X_i | MB(X_i)) \quad (8)$$

The Markov condition that a variable is independent of all other variables (except its neighbors) reduces significant computational overhead especially for large scale problems. Calculating $P(X_i | MB(X_i))$ can be done using equation (4) and equation (5). The initial states of all the variables can be chosen randomly or these can be chosen from the original small sample dataset. If the current state is $X_1 = x_1, X_2 = x_2, \dots, X_n = x_n$, then we can sample a new value x'_1 for X_1 from $P(X_1 | X_2 = x_2, \dots, X_n = x_n)$. In similar manner, we can sample the remaining new values for $X_2, X_3 \dots X_n$ until we have a new state $X_1 = x'_1, X_2 = x'_2, \dots, X_n = x'_n$. The initial samples are influenced by the

Fig. 3. Gibbs Sampling

initial distribution. At every step, we weigh our selection towards the most probable sample using the transition probability so that the samples follow the most common states accurately. Moreover, as the process is ergodic (i.e. it is possible to reach every state), it will ensure convergence to the correct distribution if sufficient number of iterations are carried out.

However, the application of Gibbs sampling for GRN estimation is somewhat limited due to the high dimensional data where number of genes is significantly higher than the samples. This means that the variance in the values taken by the variable is high, and can increase dramatically for thousands of genes and may prohibit producing independent uniform samples during sampling. The proposed new methodology for based on novel Gibbs sampling for the GRN estimation problem can overcome this limitation.

3 Methodology

The proposed MCMC sampling scheme is shown in the Fig. 4 below. In our earlier work, we employed a guided GA [21] search strategy where we had obtained a set of 10 dissimilar high scoring network structures closely representing the probability distribution using the gene expression data. With the aid of proposed methodology, we will now calculate the Bayesian posterior probability distribution of all the variables (genes) of the ten gene network structures. From the samples drawn from a network structures, we can obtain the posteriors after convergence, and then determine the state sequence and probability estimates of the model in a straight forward manner. Although the inferred high scoring network structures are disjoint (i.e. cannot be combined into one network structure), they can all be combined independently to the underlying probability distribution. Hence, all these network structures are sampled to estimate the probability distribution accurately. The important feature of our approach is the use of high scoring initial networks and a rank ordering on the network genes using Markov blankets. The convergence is obtained by running several Markov chains in parallel. Let us briefly discuss the major ‘components’ of the proposed method as they occur in Fig. 4.

Fig. 4. Proposed methodology

3.1 Rank Ordering of the Variables

As explained before, an ordinary Gibbs sampler (MCMC) chooses genes at random and then samples a new value from the estimated posterior of the neighbouring variables (i.e Markov Blanket variables). Friedman and Koller [9] argued that sampling from the space of (total) orders on variables rather than directly sampling DAGs was more efficient than application of ordinary MCMC directly in random manner. In our previous work [22], evaluating a network structure was based on the summing of scores of the individuals genes in Markov Blankets. Since the Gibbs sampler also samples the new value of a gene based on the MB variables, we will order the rank of the Markov Blankets based on their scores.

3.2 Gibbs Sampler

Before we proceed with the Gibbs sampling scheme, we need to specify a uniform prior distributions for all the genes in the domain. Rather than a random initial state of the network, we apply a standard prior which is a multivariate *Dirichlet* distribution [9]. This distribution is assigned to initial state distribution and also to the state transition distribution of the Markov chain. The initial distribution of the variables in the network (from which the initial state is sampled) is assigned using the density function estimated after smoothening of the histogram of normalized gene expression data. Sampling is straightforward as there is no evidence in the network and is done by sampling each variable in the specified rank order. For nodes without parents, sampling is done from their initial distributions while for nodes with parents, we sample from the conditional distribution of their MBs. Similarly, n independent and identically distributed samples are drawn from the target distribution $P(x)$. Since the samples drawn are continuous (through the normal range of -3 and $+3$) rather than discrete, the sampling precision is restricted to two decimal places to reduce the space of complexity. The samples collected are plotted using a histogram with n bins as shown in Fig. 4 above. The probability density function $P(x)$ of a continuous variable (gene expression) x is

approximated by smoothening of the histogram of samples as shown in the Fig. Similarly, the conditional probability distribution of all variables is estimated.

3.3 Burn-In and Convergence

The process of achieving stationary probability distribution is called as convergence while the initial phase of convergence is called the ‘burn-in’ phase. In the proposed method, the convergence is improved by running several parallel Markov chains each using a different network structure representing the probability distribution as the starting point. The idea of running multiple chain using different Bayesian network structures is mainly to obtain samples from the entire sample space of the probability distribution underlying all the structures. The chains are merged together at a certain stage of the iterations and made into a single chain.

During the process of multiple chain runs, samples are exchanged between the chains and the overall samples of a number of variables in the top of the specified order are monitored for autocorrelation and stationary distribution. A sample variation factor is introduced to determine the fraction of samples that go out of range. When the sample values do not go above a variation factor after significant number of iterations, we assume the samples have converged. From there onwards, the variable is clamped to the stationary value. This allows the sampling to be carried out on the variables that are in the lower in the rank order of the variables. In our experiments we find that the rank ordering of variables, multiple Markov chain runs and clamping also improves the mixing of samples for the unknown variables improves the mixing of samples more efficiently than an ordinary MCMC approach.

4 Experiments and Results

The validation of new techniques by comparing with other GRN reconstruction methods becomes difficult and painstaking due to non-availability of suitable benchmark dataset. Furthermore, most methods work with discrete data or perform experiments on small toy networks which also makes comparisons difficult. For this reason, in this section we validate the methods performance by investigations of synthetic datasets. Incorporating realistic relationships in synthetic data is now well established and widely used for GRN models [23]. The general idea is to obtain a synthetic network from which synthetic time course data is generated. In order to conduct tests using synthetic data set [23], several datasets are created by varying network generator parameters. To the responses of artificial GRN, generated according to topological and kinetic properties of real GRN, biological noise and technical noise is added before calculating final artificial expression ratios. The presented method is compared with a plain MCMC method which does not incorporate the improvements.

(i) Experiment 1

For the work reported in this paper, three 40 gene synthetic networks were arbitrarily generated with sample size of 100. From the set of reconstructed networks using the guided GA [21, 24] approach, we choose the first 10 high scoring networks. The probability distribution is then estimated using the proposed MCMC method. For each of the 10 structures of the networks, samples from the probability distribution

were obtained with MCMC, after discarding those from the *burn-in* phase. All simulations were repeated three times for different training data generated from synthetic networks. The results of experiments are summarized in Fig. 5. First we carry out single MCMC simulation runs instead of proposed multiple parallel MCMC runs.

From the estimated probabilities, a set of all edges whose posterior probability exceeds a given threshold $\theta \in [0, 1]$ is taken for comparison with actual network. For a given threshold θ , we count the number of true positive (*TP*), false positive (*FP*), true negative (*TN*), and false negative (*FN*) edges. We then compute the *sensitivity* = $TP / (TP + FN)$, the *specificity* = $TN / (TN + FP)$, and the complementary *specificity* = $(1 - specificity) = FP / (TN + FP)$. Rather than selecting an arbitrary value for the threshold θ , we repeat this scoring procedure for several different values of $\theta \in [0, 1]$ and plot the ensuing sensitivity scores against the corresponding complementary specificity scores. This gives the *receiver operator characteristics* (ROC) curves of Fig. 5(a). The diagonal dashed line indicates the expected ROC curve for a random predictor. The ROC curve of Fig 5(a), top left, shows that we can recover more than 80% of the true edges at approximately zero FP rate. We note that the ROC curve corresponds to the network structure obtained based on the estimated probability distribution and not based on the network reconstructed by our earlier GGA causal modeling method. The MCMC trace plot between the objective function verses cycle number for 1000 cycles for the synthetic network of 40 genes is shown in Fig 5(b). For this plot, the joint probability distribution is considered as the evaluation criterion after every run. The plot shows good mixing with a very low burn-in period. The same synthetic dataset is repeated on a plain MCMC simulation which does not incorporate the presented improvements. The trace plot of the plain MCMC simulation is shown in Fig 5(c) for 1000 cycles. It is clearly evident that mixing is poor and has a longer burn-in period. Also the simulation is oscillating around sub-optimal values of the objective function while the proposed method quickly reaches the higher values of the objective function confirming that proposed method is better than the simple MCMC. The proposed method is repeated for 500 gene synthetic network dataset and its trace plot is shown in Fig 5(d). This shows the method is easily scalable for thousands of gene as is the case of gene expression data at a comparatively feasible computational time.

With sufficient improvements identified using single MCMC runs of the presented method over the plain MCMC method, we proceed to parallel MCMC runs as presented in section 3.3. We obtained 3 different network structures for the same synthetic dataset using the GGA [21] search method and applied the network structures in a parallel MCMC runs with exchange of samples. Fig 5(e) shows the trace plot of 3 parallel MCMC runs where each chain corresponds to an individual network. From the results, it was found that the auto-correlation between the samples produced from the 3 chains was far apart during the initial 1000 samples after which the correlation increased at 2000 cycles which is an indication of convergence. The parallel runs uncovered the entire probability distribution.

Although the experiments on synthetic data are successful, the time series of 100 gene expression measurements is significantly larger than what is usually available from real world wet lab experiments; hence we also test the approach using real yeast dataset [16].

Fig. 5. Simulation results: (a) ROC plot of sensitivity versus (1-specificity) for a synthetic dataset MCMC simulation. (b) Trace plot of proposed MCMC method on synthetic network of 40 genes (c) Trace plot of plain MCMC method on synthetic network of 40 genes (d) Trace plot on synthetic network of 500 genes (e) Trace Plot of Parallel MCMC runs on 3 different network structures.

(ii) Experiment 2

The real dataset is much more complex than the synthetic data. To demonstrate the performance of the MCMC approach and also as a practical application of the method on a real biological dataset, we consider the yeast dataset [16] containing 800 genes and 77 samples comprising of a comprehensive catalogue of cell cycle-regulated genes in the yeast *Saccharomyces cerevisiae*. The dataset includes three long time course expression values representing three different ways of synchronizing the normal cell cycle, and five shorter time courses representing the altered environmental conditions. These results were combined with those by Cho et al.,[25] to produce a

more comprehensive collection of data. The test samples were synchronized so that all the cells would be at the same stage in their cell cycle. Using this data set, gene networks have already been reconstructed (Friedman et al., 2001). We also note that the Spellman dataset has classified the 800 genes in different phases of cell cycle such as G1, G2, S and M/G1 and G2/M. Using the MCMC based probability inference; the minimal I-map of the inferred yeast network is obtained. It is shown in Fig. 6 below.

Fig. 6. Yeast Gene Regulatory network Minimal I-map

From the reconstructed network shown in Fig. 6, we note the following interactions taking place which is in confirmation with the available literature [26].

1. MBF (a complex of MBP1 and SWI6) and SBF (a complex of SWI4 and SWI6) controls the late G1 genes (e.g. CLN2 and NDD1).
2. MCM1, together with FKH1 or FKH2, recruits the NDD1 protein in late G2 and controls the transcription of G2/M genes.
3. SWI5 and ACE2 regulate genes at M/G1.
4. MCM1 regulates SIC1 and ACE2.

5 Conclusion

A new Markov chain Monte Carlo approach using Gibbs sampling is presented for estimating the conditional probability distribution underlying gene regulatory network structures. The approach is novel as it performs the rank ordering of genes based on the Markov Blanket scoring metric, applies parallel Markov chains using different high scoring starting network structures and clamps genes which are higher in the order for faster and efficient convergence. Rather than initializing the Markov chains with randomly chosen networks, our previously reported guided GA is used to generate the high scoring initial networks and the probability distribution. Both synthetic and real world yeast cell data sets have been applied in the investigations. The experiment on synthetic data set that the proposed technique performs significant better than the standard MCMC algorithm for estimating probability distributions of the genes in the network. From the yeast cell cycle data experiment, we observe that the

minimal network derived using real life yeast dataset has more accurate reconstruction of regulatory interactions. However, due to the nature of the microarray data set, the resulting minimal GRN is not unique. With integration of other related data such as sequence analysis in the form of prior probability, it would be worth investigating to recover unique minimal network which represents the underlying structure of gene expression. This is currently under investigations.

References

1. D'haseleer, P., Liang, S., Somogyi, R.: Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* 16, 707–726 (2000)
2. Heckerman, D.: A Tutorial on Learning with Bayesian Networks. In: Jordan, M. (ed.) *Learning in Graphical Models*. MIT Press, Cambridge (1999)
3. Pearl, J.: *Causality: Models, Reasoning and Inference*. Cambridge University Press, Cambridge (2000)
4. Sprites, P., Glymour, C., Scheines, R.: *Causation, Prediction, and Search: Adaptive Computation and Machine Learning*, 2nd edn. MIT Press, Cambridge (2001)
5. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian networks to analyze expression data. *J. Comp. Biol.* 7, 601–620 (2000)
6. Friedman, N.: Inferring cellular network using probabilistic graphical models. *Science* 33, 799–805 (2004)
7. Nachman, I., Friedman, N.: Inferring quantitative models of regulatory networks from expression data. *Bioinformatics* 20, I248–I256 (2004)
8. Chickering, D.M.: Learning Equivalence Classes of Bayesian-Network Structures. *Journal of Machine Learning Research* 2, 445–498 (2002)
9. Friedman, N., Koller, D.: Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning* 50, 95–126 (2003)
10. de Hoon, S.I., Kobayashi, K., Ogasawara, N., Miyano, S.: Inferring Gene Regulatory Networks From Time-Ordered Gene Expression Data Of *Bacillus Subtilis* Using Differential Equations. In: *Pacific symposium on computation biology*, vol. 8, pp. 17–28 (2003)
11. Murphy, K., Mian, S.: Modelling gene expression data using dynamic Bayesian networks, in *Technical Report*. University of California, Berkeley (1999)
12. Madigan, D., Andersson, S., Perlman, M., Volinsky, C.: Bayesian model averaging and model selection for Markov equivalence classes of acyclic graphs. *Communications in Statistics: Theory and Methods* 25, 2493–2519 (1996)
13. Madigan, D.a.J.Y.: Bayesian graphical models for discrete data. *International statistical Review* 63, 215–232 (1995)
14. Giudici, P.a.P.G.: Decomposable graphical Gaussian model determination. *Biometrika* 86(4), 785–801 (1999)
15. Giudici, P., Green, P., Tarantola, C.: Efficient model determination for discrete graphical models, in *Technical Report*. Univ. Pavia, Italy (2000)
16. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell.* 9, 3273–3297 (1998)
17. Liu, J.S.: *Monte Carlo Strategies in Scientific Computing*. Springer, Heidelberg (2001)
18. Henrion, M.: Practical issues in constructing a Bayes belief network. *Int. J. Approx. Reasoning* 2(3), 337 (1988)

19. Gilks, W.R., Wild, P.: Adaptive Rejective Sampling for Gibbs Sampling. *Applied Statistics* 41, 337–348 (1992)
20. Shachter, R.D., Peot, M.A.: Simulation approaches to general probabilistic inference on belief networks. In: *Uncertainty in Artificial Intelligence*, vol. 5, pp. 221–231 (1989)
21. Ram, R., Chetty, M.: A Guided genetic algorithm for Gene Regulatory Network. In: *IEEE Congress on Evolutionary Computation*, Singapore (2007)
22. Ram, R., Chetty, M.: Constraint Minimization for Efficient Modeling of Gene Regulatory Network. In: Chetty, M., Ngom, A., Ahmad, S. (eds.) *PRIB 2008. LNCS (LNBI)*, vol. 5265, pp. 201–213. Springer, Heidelberg (2008)
23. Ram, R., Chetty, M.: Generating Synthetic Gene Regulatory Networks. In: Chetty, M., Ngom, A., Ahmad, S. (eds.) *PRIB 2008. LNCS (LNBI)*, vol. 5265, pp. 237–249. Springer, Heidelberg (2008)
24. Ram, R., Chetty, M.: A Markov blanket based Probabilistic Genetic Algorithm for Causal Reconstruction of Gene Regulatory Networks. *BioSystems Special Issue on Evolving Gene Regulatory Networks* (submitted, 2009)
25. Cho, R.J., et al.: A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell* 2, 65–73 (1998)
26. Chen, K.C., Calzone, L., Csikasz-Nagy, A., Cross, F.R., Novak, B., Tyson, J.J.: Integrative analysis of cell cycle control in budding yeast. *Mol. Biol. Cell* 15, 3841–3862 (2004)