TECHNICAL REPORT

A Polynomial Time Algorithm for Learning Globally Optimal Dynamic Bayesian Network

and

its Applications in Genetic Network Reconstruction

FIT-GSIT TR.1101

VINH NGUYEN [1], MADHU CHETTY [1], ROSS COPPEL [2] AND PRAMOD P. WANGIKAR [3]

[1] Faculty of Information Technology, Monash University
[2] Faculty of Medicine, Nursing and Health Sciences, Monash University
[3] Chemical Engineering Department, Indian Institute of Technology, Bombay

{Vinh.Nguyen,Madhu.Chetty,Ross.Coppel}@Monash.edu
Wangikar@iitb.ac.in

Contents

One

GlobalMIT—A Polynomial Time Algorithm for Learning Globally Optimal Dynamic Bayesian Network

The work in this chapter is concerned with the problem of learning the globally optimal structure of a dynamic Bayesian network (DBN) from data. We propose using a recently introduced information theoretic criterion named MIT (Mutual Information Test) for evaluating the goodness-of-fit of a DBN structure. MIT has been previously shown to be effective for learning static Bayesian network, yielding results competitive to other popular scoring metrics, such as BIC/MDL, K2 and BD, and the well-known constraint-based approach PC algorithm. This work adapts MIT to the case of DBN. Using a modified variant of MIT, we show that learning the globally optimal DBN structure can be efficiently achieved in polynomial time.

1.1 Introduction

Bayesian network (BN) is a central topic in machine learning, and has found applications in various fields. A (static) BN is defined by a graphical structure and a family of probabilistic distribution, which together allow efficient and accurate representation of the joint probability distribution of a set of random variables (RV) of interest [KF09]. The graphical part of a static BN is a directed acyclic graph (DAG), with nodes representing RVs and edges representing their conditional (in)dependence relations.

Two important disadvantages when applying static BN to certain domain problems, such as gene regulatory network reconstruction in bioinformatics, are that: *(i)* BN does not have a mechanism for exploiting the temporal aspect of time-series data (such as that gathered from time-series microarray experiments) abundant in this field; and *(ii)* BN does not allow the modeling of cyclic phenomena, such as feed back loops, which are prevalent in biological systems [YSW+04]. These drawbacks have motivated the development of the so-called dynamic Bayesian network (DBN). The simplest model of this type is the first-order Markov stationary DBN, in which both the structure of the network and the parameters characterizing it are assumed to remain unchanged over time, such as the one exemplified in Figure 1.1a. In this model, the value of a RV at time $t + 1$ is assumed to depend only on the value of its parents at time t. DBN accounts for the temporal aspect of time-series data, in that an edge must always direct forward in time (i.e., cause must precede consequence), and allows feedback loops (Fig. 1.1b). Since its inception, DBN has received particular interest, especially from the bioinformatics community [MM99, PRM+03, KIM03, Hus03, YSW+04, ZC05, WD09]. Recent works in the machine learning

community have progressed to allow more flexible DBN models, such as one with, either parameters [GH09], or both structure and parameters [RH09, RH10, DLH10] changing over time. It is worth noting that more flexible models generally require more data to be learned accurately. In situations where training data are scarce, such as in microarray experiments where the data size can be as small as a couple of dozen samples, a simpler model such as the first-order Markov stationary DBN might be a more suitable choice.

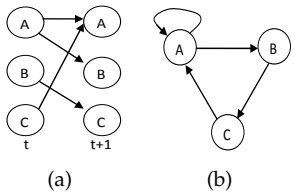


Figure 1.1: Dynamic Bayesian Network: (a) a 1st order Markov stationary DBN; (b) its equivalent folded network

In this work, we focus on the problem of learning the globally optimal structure for the first-order Markov stationary DBN. Henceforth, DBN shall refer to this particular class of stationary DBN, and *learning* shall refer to *structure learning*. The most popular approaches for learning DBN have been the ones adapted from the static BN literature, namely the *search+score* paradigm [YSW+04, WD09], and Markov Chain Monte Carlo (MCMC) simulation [Hus03, DLH10, RH10]. Herein we are interested in the *search+score* approach, in which one has to specify a scoring function to assess the goodness-of-fit of a DBN given the data, and a search procedure to find the optimal network based on this scoring metric. Several popular scores for static BN, such as the Bayesian scores (K2, Bayesian-Dirichlet (BD), BDe and BDeu), and the information theoretic scores (Bayesian Information Criterion (BIC)/minimal description length (MDL) and Akaike Information Criterion—AIC), can be adapted straightforwardly for DBN. Another recently introduced scoring metric that catches our interest is the so-called MIT (Mutual Information Test) score [dC06], which, as the name suggests, belongs to the family of scores based on information theory. Through extensive experimental validation, the author suggested that MIT can compete favorably with Bayesian scores, outperforms BIC/MDL and should be the score of reference within those based on information theory. As opposed to the other popular scoring metrics, MIT has not been considered for DBN learning to our knowledge.

As for the *search* part, due to several non-encouraging complexity results in learning static BN, most authors have resorted to heuristic search algorithms when it comes to learning DBN. For example, Chickering [Chi96] showed that under the BDe metric, the problem of identifying a Bayesian network, with each node having at most K parents, that has a relative posterior probability greater than a given constant is NP-complete, even with $K = 2$. For this reason, greedy hill-climbing and meta-optimization frameworks, such as Tabu search, simulated annealing and genetic algorithms are often employed. In a recent publication, Dojer [Doj06] has shown otherwise that learning DBN structure, as opposed to static BN, does not necessarily have to be NP-hard. In particular, this author showed that, under some mild assumptions, there are algorithms for finding the globally optimal network with a polynomial worst-case time complexity, when the MDL and BDe scores

are used. In the same line of these findings, in this work, we shall show that there exists a polynomial worst-case time complexity algorithm for learning the globally optimal DBN under the newly introduced MIT scoring metric, which we shall name the globalMIT algorithm. Our experimental results show that, in terms of the recovered DBN quality, MIT performs competitively with BIC/MDL and BDe. In terms of theoretical complexity analysis, globalMIT admits a comparable worst-case complexity to the BIC/MDL-based global algorithm, and is much faster than the BDe-based algorithm.

The chapter is organized as follows: in Section 1.2 we review the MIT score for DBN learning. Section 1.3 presents our algorithm for finding the globally optimal network, followed by experimental results in section 1.4, and finally some discussion and conclusion.

1.2 MIT Score for Dynamic Bayesian Network Structure Learning

In this section, we review the MIT score for learning BN, then adapts it to the DBN case. Briefly speaking, under MIT the goodness-of-fit of a network is measured by the total mutual information shared between each node and its parents, penalized by a term which quantifies the degree of statistical significance of this shared information. Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ denote the set of n variables with corresponding $\{r_1, \ldots, r_n\}$ discrete states, D denote our data set of N observations, G be a DAG, and $\mathbf{Pa}_i = \{X_{i1}, \ldots, X_{is_i}\}$ be the set of parents of X_i in G with corresponding $\{r_{i1}, \ldots, r_{is_i}\}$ discrete states, $s_i = |\mathbf{Pa}_i|$, then the MIT score is defined as:

$$SS_{MIT}(G : D) = \sum_{\substack{i=1 \\ \mathbf{Pa}_i \neq \emptyset}}^{n} \{2N.I(X_i, \mathbf{Pa}_i) - \sum_{j=1}^{s_i} \chi_{\alpha, l_i\sigma_i(j)}\}$$

where $I(X_i, \mathbf{Pa}_i)$ is the mutual information between X_i and its parents as estimated from D. $\chi_{\alpha,l_{ij}}$ is the value such that $p(\chi^2(l_{ij}) \leq \chi_{\alpha,l_{ij}}) = \alpha$ (the Chi-square distribution at significance level $1 - \alpha$), and the term $l_i\sigma_i(j)$ is defined as:

$$l_{i\sigma_i(j)} = \begin{cases} (r_i - 1)(r_{i\sigma_i(j)} - 1) \prod_{k=1}^{j-1} r_{i\sigma_i(k)}, & j = 2 \ldots, s_i \\ (r_i - 1)(r_{i\sigma_i(j)} - 1), & j = 1 \end{cases}$$

where $\sigma_i = \{\sigma_i(1), \ldots, \sigma_i(s_i)\}$ is any permutation of the index set $\{1 \ldots s_i\}$ of $\mathbf{Pa}_i$, with the first variable having the greatest number of states, the second variable having the second largest number of states, and so on.

To make sense of this criterion, let us first point out that maximizing the first term in the score, $\sum_{\substack{i=1 \\ \mathbf{Pa}_i \neq \emptyset}}^{n} 2N.I(X_i, \mathbf{Pa}_i)$, can be shown to be equivalent to maximizing the log-likelihood criterion. Learning BN by using the maximum likelihood principle suffers from overfitting however, as the fully-connected network will always have the maximum likelihood. Likewise, for the MIT criterion, since the mutual information can always be increased by including additional variables to the parent set, i.e., $I(X_i, \mathbf{Pa}_i \cup X_j) \geq I(X_i, \mathbf{Pa}_i)$, the complete network will have the maximum total mutual information.

Thus, there is a need to penalize the complexity of the learned network. Penalizing the log-likelihood criterion with $-\frac{1}{2}C(G)\log(N)$ gives us the BIC/MDL criteria, while $-C(G)$ gives us the AIC criterion (where $C(G) = \sum_{i=1}^{n}(r_i - 1)\prod_{j=1}^{s_i} r_{ij}$ measures the network complexity). As for the MIT criterion, while the mutual information always increases

when including additional variables to the parent set, the degree of statistical significance of this increment might become negligible as more and more variables are added. This significance degree can be quantified based on a classical result in information theory by Kullback [Kul68], which, in this context, can be stated as follows: under the hypothesis that X_i and X_j are conditionally independent given $\mathbf{Pa}_i$ is true, the statistics $2N.I(X_i, X_j|\mathbf{Pa}_i)$ approximates to a $\chi^2(l)$ distribution, with $l = (r_i - 1)(r_j - 1)q_i$ degree of freedom, and $q_i = 1$ if $\mathbf{Pa}_i = \emptyset$, otherwise q_i is total the number of state of $\mathbf{Pa}_i$, i.e., $q_i = \prod_{k=1}^{s_i} r_{ik}$. Now we can see that the second term in the MIT score penalizes the addition of more variables to the parent set. Roughly speaking, only variables that have the conditional mutual information shared with X_i given all the other variables in $\mathbf{Pa}_i$ that is higher than 100α percent of the MI values under the null hypothesis of independence can increase the score. For detailed motivations and derivation of this scoring metric as well as an extensive comparison with BIC/MDL and BD, we refer readers to [dC06].

Adapting MIT for DBN learning is rather straightforward. One just has to pay attention to the fact that the mutual information is now calculated between a parent set and its child, which should be 1-unit shifted in time, as required by the first-order Markov assumption, denoted by $X_i^{\overrightarrow{1}} = \{X_{i2}, X_{i3}, \ldots, X_{iN}\}$. As such, the number of "effective" observations, denoted by N_e, for DBN is now only $N - 1$. This is demonstrated in Figure 1.2. The MIT score for DBN should be calculated as:

$$S'_{MIT}(G : D) = \sum_{\substack{i=1 \\ \mathbf{Pa}_i \neq \emptyset}}^{n} \{2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{Pa}_i) - \sum_{j=1}^{s_i} \chi_{\alpha, l_i \sigma_i(j)}\}$$

Similarly, when the data is composed of N_t separate time-series, the number of effective observations is only $N_e = N - N_t$.

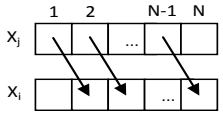


Figure 1.2: Data alignment for dynamic Bayesian network with an edge $X_j \rightarrow X_i$. The "effective" number of observations is now only $N - 1$.

1.3 Optimal Dynamic Bayesian Network Structure Learning in Polynomial Time with MIT

In this section, we show that learning the globally optimal DBN with MIT can be achieved in polynomial time. Our development is based on a recent result presented in [Doj06], which states that under several mild assumptions, there exists a polynomial worst-case time complexity algorithm for learning the optimal DBN with the MDL and BDe scoring metrics. Specifically, the 4 assumptions that Dojer considered are:

Assumption 1. *(acyclicity) There is no need to examine the acyclicity of the graph.*

Assumption 2. *(additivity)* $S(G : D) = \sum_{i=1}^{n} s(X_i, \boldsymbol{Pa}_i : D|_{X_i \cup \boldsymbol{Pa}_i})$ *where* $D|_{X_i \cup \boldsymbol{Pa}_i}$ *denotes the restriction of* D *to the values of the members of* $X_i \cup \boldsymbol{Pa}_i$.

To simplify notation, we write $s(\mathbf{Pa}_i)$ for $s(X_i, \mathbf{Pa}_i : D|_{X_i \cup \mathbf{Pa}_i})$.

Assumption 3. *(splitting)* $s(\boldsymbol{Pa}_i) = g(\boldsymbol{Pa}_i) + d(\boldsymbol{Pa}_i)$ *for some non-negative functions g,d satisfying* $\boldsymbol{Pa}_i \subseteq \boldsymbol{Pa}'_i \Rightarrow g(\boldsymbol{Pa}_i) \leq g(\boldsymbol{Pa}'_i)$

Assumption 4. *(uniformity)* $|\boldsymbol{Pa}_i| = |\boldsymbol{Pa}'_i| \Rightarrow g(\boldsymbol{Pa}_i) = g(\boldsymbol{Pa}'_i)$

Assumption 1 is valid for DBN in general. For the first-order Markov DBN that we are considering in this work, since the graph is bipartite, with edges directing only forward in time (Fig. 1.1a), acyclicity is automatically satisfied. Assumption 2 simply states that the scoring function decomposes over the variables, which is satisfied by most scoring metrics such as BIC/MDL, BD and also clearly by MIT. Together with assumption 1, this assumption allows us to compute the parents set of each variable independently. Assumption 3 requires the scoring function to decompose into two components: d evaluating the accuracy of representing the distribution underlying the data by the network, and g measuring its complexity. Furthermore, g is required to be a monotonically non-decreasing function in the cardinality of $\mathbf{Pa}_i$ (assumption 4), i.e., the network gets more complex as more variables are added to the parent sets.

We note that unlike MIT in its original form that we have considered above, where better networks have higher scores, for the score considered by Dojer, lower scored networks are better. And thus the corresponding optimization must be cast as a score minimization problem. We now consider a variant of MIT as follows:

$$S_{MIT}(G : D) = \sum_{i=1}^{n} 2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{X}) - S'_{MIT}(G : D) \tag{1.1}$$

which admits the following decomposition over each variable (with the convention of $I(X_i, \emptyset) = 0$):

$$\begin{aligned} s_{MIT}(\mathbf{Pa}_i) &= d_{MIT}(\mathbf{Pa}_i) + g_{MIT}(\mathbf{Pa}_i) \\ d_{MIT}(\mathbf{Pa}_i) &= 2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{X}) - 2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{Pa}_i) \\ g_{MIT}(\mathbf{Pa}_i) &= \sum_{j=1}^{s_i} \chi_{\alpha, l_i \sigma_i(j)} \end{aligned}$$

Roughly speaking, d_{MIT} measures the "error" of representing the joint distribution underlying D by G, while g_{MIT} measures the complexity of this representation. We state the following results:

Proposition 1. *The problem of* S'_{MIT} *maximization is equivalent to the problem of* S_{MIT} *minimization.*

Proof. Obvious, since $\sum_{i=1}^{n} 2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{X}) = const.$ □

Proposition 2. d_{MIT}, g_{MIT} *satisfy assumption 3*

Proof. $d_{MIT} \geq 0$ since of all parent sets $\mathbf{Pa}_i$, $\mathbf{X}$ has the maximum mutual information with $X_i^{\overrightarrow{1}}$. And since the support of the Chi-square distribution is $\mathbb{R}^+$, i.e., $\chi_{\alpha,\cdot} \geq 0$, therefore $\mathbf{Pa}_i \subseteq \mathbf{Pa}'_i \Rightarrow 0 \leq g_{MIT}(\mathbf{Pa}_i) \leq g_{MIT}(\mathbf{Pa}'_i)$. □

Unfortunately, g_{MIT} does not satisfy assumption 4. However, for many applications, if all the variables have the same number of states then it can be shown that g_{MIT} satisfies assumption 4.

Assumption 5. *(variable uniformity) All variables in $\mathbf{X}$ have the same number of discrete states k.*

Proposition 3. *Under the assumption of variable uniformity, g_{MIT} satisfies assumption 4.*

Proof. It can be seen that if $|\mathbf{Pa}_i| = |\mathbf{Pa}'_i| = s_i$, then $g_{MIT}(\mathbf{Pa}_i) = g_{MIT}(\mathbf{Pa}'_i) = \sum_{j=1}^{s_i} \chi_{\alpha,(k-1)^2 k^{j-1}}$. □

Since $g_{MIT}(\mathbf{Pa}_i)$ is the same for all parent sets of the same cardinality, we can write $g_{MIT}(|\mathbf{Pa}_i|)$ in place of $g_{MIT}(\mathbf{Pa}_i)$. With assumptions 1-5 satisfied, we can employ the following Algorithm 1, named globalMIT, to find the globally optimal DBN with MIT, i.e., the one with the minimal S_{MIT} score.

Algorithm 1 globalMIT : Optimal DBN with MIT

$\mathbf{Pa}_i := \emptyset$
for $p = 1$ **to** n **do**
  If $g_{MIT}(p) \geq s_{MIT}(\mathbf{Pa}_i)$ then return $\mathbf{Pa}_i$; Stop.
  $\mathbf{P} = \arg\min_{\{\mathbf{Y} \subseteq \mathbf{X}: |\mathbf{Y}| = p\}} s_{MIT}(\mathbf{Y})$
  If $s_{MIT}(\mathbf{P}) < s_{MIT}(\mathbf{Pa}_i)$ then $\mathbf{Pa}_i := \mathbf{P}$.
end for

Theorem 1. *Under assumptions 1-5, globalMIT applied to each variable in $\mathbf{X}$ finds a globally optimal DBN under the MIT scoring metric.*

Proof. The key insight here is that once a parent set grows to a certain extent, its complexity alone surpasses the total score of a previously found sub-optimal parent set. In fact, all the remaining potential parent sets $\mathbf{P}$ omitted by the algorithm have a total score higher than the current best score, i.e., $s_{MIT}(\mathbf{P}) \geq g_{MIT}(|\mathbf{P}|) \geq s_{MIT}(\mathbf{Pa}_i)$, where $\mathbf{Pa}_i$ is the last sub-optimal parent set found. □

We note that the terms $2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{X})$ in the S_{MIT} score in (1.1) do not play any essential role, since they are all constant and would not affect the outcome of our optimization problem. Knowing their exact value is however, necessary for the stopping criterion in Algorithm 1, and also for constructing its complexity bound, as we shall do shortly. Unfortunately, calculating $I(X_i^{\overrightarrow{1}}, \mathbf{X})$ is by itself a hard problem, requiring $O(k^{n+1})$ space and time in general. However, for our purpose, since the only requirement for d_{MIT} is that it must be non-negative, it is sufficient to use an upper bound of $I(X_i^{\overrightarrow{1}}, \mathbf{X})$. A fundamental property of the mutual information states that $I(\mathbf{X}, \mathbf{Y}) \leq \min\{H(\mathbf{X}), H(\mathbf{Y})\}$, i.e., mutual information is bounded by the corresponding entropies. We therefore have:

$$2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{X}) \leq 2N_e.H(X_i^{\overrightarrow{1}}),$$

where $H(X_i^{\overrightarrow{1}})$ can be estimated straightforwardly from the data. Or else, we can use an a priory fixed upper bound for all $H(X_i^{\overrightarrow{1}})$, that is $\log k$, then:

$$2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{X}) \leq 2N_e.\log k.$$

Using these bounds, we obtain the following more practical versions of d_{MIT}:

$$\begin{aligned} d'_{MIT}(\mathbf{Pa}_i) &= 2N_e.H(X_i^{\overrightarrow{1}}) - 2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{Pa}_i) \\ d''_{MIT}(\mathbf{Pa}_i) &= 2N_e.\log k - 2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{Pa}_i) \end{aligned}$$

It is straightforward to show that Algorithm 1 and Theorem 1 are still valid when d'_{MIT} or d''_{MIT} are used in place of d_{MIT}.

1.3.1 Complexity bound

Theorem 2. *globalMIT admits a polynomial worst-case time complexity in the number of variables.*

Proof. Our aim is to find a number p^* satisfying $g_{MIT}(p^*) \geq s_{MIT}(\emptyset)$. Clearly, there is no need to examine any parent set of cardinality p^* and over. In the worse case, our algorithm will have to examine all the possible parent sets of cardinality from 1 to $p^* - 1$. We have:

$$\begin{aligned} g_{MIT}(p^*) &\geq s_{MIT}(\emptyset) \\ \Leftrightarrow \sum_{j=1}^{p^*} \chi_{\alpha, l_i \sigma_i(j)} &\geq d_{MIT}(\emptyset) = 2N_e.I(X_i^{\overrightarrow{1}}, \mathbf{X}). \end{aligned}$$

As discussed above, since calculating d_{MIT} is not convenient, we use d'_{MIT} and d''_{MIT} instead. With d'_{MIT} p^* can be found as:

$$p^* = \underset{\sum_{j=1}^{p} \chi_{\alpha, l_i \sigma_i(j)} \geq 2N_e.H(X_i^{\overrightarrow{1}})}{\arg\min} p, \tag{1.2}$$

while with d''_{MIT}:

$$p^* = \underset{\sum_{j=1}^{p} \chi_{\alpha, l_i \sigma_i(j)} \geq 2N_e.\log k}{\arg\min} p. \tag{1.3}$$

It can be seen that p^* depends only on α, k and N_e. Since there are $O(n^{p^*})$ subsets with at most p^* parents, and each set of parents can be scored in polynomial time, globalMIT admits an overall polynomial worst-case time complexity in the number of variable n. □

We now give some examples to demonstrate the practicability of Theorem 2.

Example 1: Consider a gene regulatory network reconstruction problem, where each gene has been discretized to $k = 3$ states, corresponding to up, down and regular gene expression. With the level of significance α set to 0.999 as recommended in [dC06], we have:

$$
\begin{aligned}
g_{MIT}(1) &= \chi_{0.999,4} &&= 18.47\\
g_{MIT}(2) &= g_{MIT}(1) + \chi_{0.999,12} &&= 51.37\\
g_{MIT}(3) &= g_{MIT}(2) + \chi_{0.999,36} &&= 119.35\\
g_{MIT}(4) &= g_{MIT}(3) + \chi_{0.999,108} &&= 278.51\\
g_{MIT}(5) &= g_{MIT}(4) + \chi_{0.999,324} &&= 686.92\\
g_{MIT}(6) &= g_{MIT}(5) + \chi_{0.999,972} &&= 1800.9\\
g_{MIT}(7) &= g_{MIT}(6) + \chi_{0.999,2916} &&= 4958.6\\
g_{MIT}(8) &= g_{MIT}(7) + \chi_{0.999,8748} &&= 14121\\
g_{MIT}(9) &= g_{MIT}(8) + \chi_{0.999,26244} &&= 41079\\
g_{MIT}(10) &= g_{MIT}(9) + \chi_{0.999,78732} &&= 121042\\
&\dots
\end{aligned}
$$

Consider a data set of $N = 12$ observations, which is the popular length of microarray time-series experiments (in fact N often ranges within $4 - 15$), then $d''_{MIT}(\emptyset) = 2(N - 1)\log k = 24.16$. Observing that $g_{MIT}(2) > d''_{MIT}(\emptyset)$, then $p^* = 2$ and we do not have to consider any parent sets of 2 variables or more.

Let us compare this bound with those of the algorithms for learning the globally optimal DBN under the BIC/MDL and BDe scoring metrics. For BIC/MDL, p^*_{MDL} is given by $\lceil \log_k N \rceil$, while for BDe, $p^*_{BDe} = \lceil N \log_{\gamma^{-1}} k \rceil$, where the distribution $P(G) \propto \gamma^{\sum |\mathbf{Pa}_i|}$, with a penalty parameter $0 < \gamma < 1$, is used as a prior over the network structures [Doj06]. In this case, $p^*_{MDL} = 3$. If we choose $\log \gamma^{-1} = 1$ then $p^*_{BDe} = \lceil N \log k \rceil = 14$. In general, p^*_{BDe} scales linearly with the number of data items N, making its value less of practical interest, even for small data sets.

Example 2: Since the number of observations in a single microarray time-series experiment is often limited, it is a popular practice to concatenate several time-series to obtain a larger data set for analysis. Let us merge $N_t = 10$ data sets, each with 12 observations, then $N_e = N - N_t = 120 - 10 = 110$. For this combined data set, $g_{MIT}(4) > d''_{MIT}(\emptyset) = 2N_e \log k = 241.69 \Rightarrow p^* = 4$, thus there is no need to consider any parent set of more than 3 variables. For comparison, we have $p^*_{MDL} = 5$, and $p^*_{BDe} = 132$ with $\log \gamma^{-1} = 1$.

Of course, this analysis only gives us the worst-case time complexity. In practice, the execution of Algorithm 1 can often be much shorter, since $s_{MIT}(\mathbf{Pa}_i)$ is often much greater than $s_{MIT}(\emptyset)$. This observation also applies for the global algorithms based on the BIC/MDL and BDe scoring metrics.

Even though Algorithm 1 admits a polynomial time complexity, exhaustive search for the optimal parent sets over all the subsets of $\mathbf{X}$ with at most $p^* - 1$ elements may still be extremely time consuming, especially when the number of variable n is large. In such cases, MIT may be instead used in conjunction with local or meta-optimization algorithms such as hill climbing, simulated annealing or genetic algorithm. In these cases, our analysis gives a theoretical guidance and justification for setting the so-called "max-fan-in" parameter, which dictates the maximum number of parents allowed for each node, as found popular in many softwares for DBN learning. Setting an appropriate max-fan-in number greatly reduces the search space, while still ensuring that the region containing the global

optimum is covered. There seems to be no systematic rules for setting this parameter in the BN literature in our observation.

Example 3: Let's consider some large scale data sets, with $k = 3, \alpha = 0.999$ and a set of $N = 10000$ observations, then $p^* = 9$. The max-fan-in parameter can then be set to 8. For comparison, we have $p^*_{MDL} = 9$ and $p^*_{BDE} = 10987$ with $\log \gamma^{-1} = 1$.

1.3.2 Efficient Implementation for globalMIT

The search procedure involves examining all potential parent sets of increasing cardinality. The following decomposition property of the mutual information is handy when it comes to design an efficient implementation for globalMIT:

$$I(X_i, \mathbf{Pa}_i \cup X_j) = I(X_i, \mathbf{Pa}_i) + I(X_i, X_j | \mathbf{Pa}_i)$$

This implies that the mutual information can be computed incrementally, and suggests that, for efficiency, the computed mutual information values should be cached to avoid redundant computations, subject to memory availability.

1.4 Experimental Evaluation

In this section, we describe our experiments to evaluate our global approach for learning DBN with MIT, and compare it with the other most popular scores, namely BIC/MDL and BD. Our method, implemented in Matlab, was used, along with BNFinder [WD09], a Python-based software for inferring the globally optimal DBN with the MDL and BDe scores as proposed by Dojer [Doj06]. In addition, we also employed the Java-based Banjo software [Har], which can perform greedy search and simulated annealing over the DBN space using the BDeu metric. BNFinder and Banjo are freely available online.

Data: The specific problem domain that we shall work with in this experiment is the problem of gene regulatory network reconstruction from microarray data, with the variables being genes, and edges being regulatory relationship between genes. The experiments are carried out on several synthetic data sets with known ground-truth network. In order to eliminate the bias toward a certain data generation method, we shall employ several different data generation schemes that have been used in some previous studies, namely, probabilistic method [Hus03], linear dynamical system based method [YSW+04], and non-linear dynamical system based method [SI04]. As a realistic number of samples for microarray data, we generated data sets of between 30 and 300 samples.

Data preprocessing: As per requirement of assumption 5, and as per popular practice in the field, all the gene expression profiles were discretized to the same number of levels (normally 3). As mentioned in section 1.2, data that consist of multiple separate time series need to be preprocessed to have the proper data alignment and the number of effective samples. BNFinder has an in-built capability to process multiple time series, while Banjo doesn't seem to support this functionality. However, when the number of samples is large, this preprocessing will probably only marginally affect the results.

Self-regulated links handling: During the experiments, we have noted that, the self-regulated links, i.e., a link from a node to itself, very frequently appear in the optimal networks. This is to be expected, since biological time series data are often smooth and have a high degree of auto-correlation and auto-mutual information at short lag. While these links might or might not be present in the ground-truth network, they are generally

not very informative, as for most natural biological processes, the current state often dictates the state in the near future (unless the process evolves in a random-walk manner). Different softwares have different policies with these self-regulated links: BNFinder suppresses these links by default, i.e., the search is performed only on the parent sets without self-parents, while Banjo on the other hand, has a parameter to enforce this type of link, but then simply ignores them all in its output report. For globalMIT, we also implemented an option to either enable or disable these links. Self-links are, however, not considered when calculating the performance metrics below.

Performance metrics: With the ground-truth network available, we count the number of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) edges, and report two network quality metrics, namely *sensitivity*= TP/(TP+FN), and *imprecision*=FP/(FP+TP).

Parameters setting: globalMIT has one parameter, namely the significance level α, to control the trade-off between goodness-of-fit and network complexity. Adjusting α will generally affect the sensitivity and imprecision of the discovered network, very much like its affect on the Type-I and Type-II error of the mutual information test of independence. de Campos [dC06] suggested using very high levels of significance, namely 0.999 and 0.9999. We note that, the data sets used in [dC06] are of sizes from 1000 to 10000 samples. For microarray data sets of merely $30 - 300$ samples, it is necessary to use a lower level of significance α to avoid overly penalizing network complexity. We have experimentally observed that using $\alpha \in [0.95, 0.999]$ on these small data sets yielded reasonable results, with balanced sensitivity and imprecision.

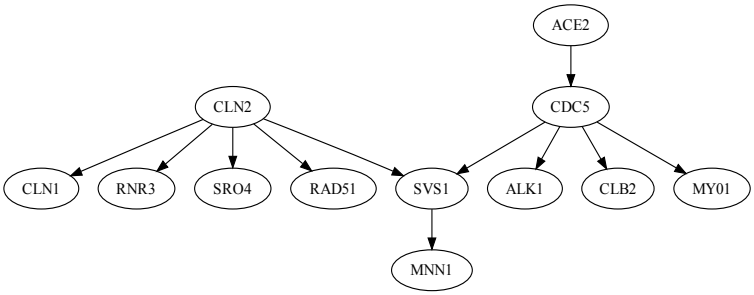
BNFinder+MDL required no parameter tuning, while for BNFinder+BDe, the pseudocounts for the BDe score was set to the default value of 1, and the penalty parameter was set to the default value of $\log \gamma^{-1} = 1$. For Banjo, we employed simulated annealing as the search engine, and left the equivalent sample size to the default value of 1 for the BDeu score, while the max-fan-in was set to 3. The runtime for Banjo was set to the average runtime of globalMIT, with a minimum value of 10 minutes, in case where globalMIT terminates earlier. Since some experiments were time consuming, all our experiments were performed in parallel on a 16-core Xeon X5550 workstation.

1.4.1 Results on Probabilistic Network Synthetic Data

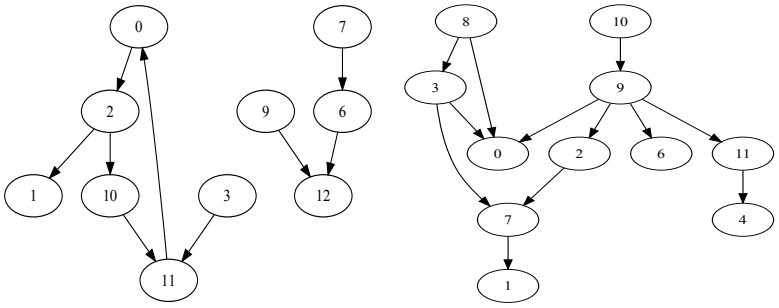
We employed a subnetwork of the yeast cell cycle, consisting of 12 genes and 11 interactions, as depicted in Fig. 1.3(a). Two different conditional probabilities were associated with these interactions, namely noisy regulation according to a binomial distribution, and noisy XOR-style co-regulation (see [Hus03] for the parameter details, and this author website for Matlab code to generate this data). In addition, 8 unconnected nodes were added as confounders, for a total of 20 nodes. For each number of samples $N = 30, 70$ and 100, we generated 10 data sets. From the average statistics in Table 2.1, it can be seen that this is a relatively easy case for all methods. Except Banjo which committed a lower sensitivity and yet a higher imprecision, all other methods nearly recovered the correct network. Note that due to the excessive runtime of BNFinder+BDe, for $N = 100$, only 1 of ten data sets was analyzed.

1.4.2 Results on Linear Dynamical System Synthetic Data

We employed the two synthetic networks as described in [YSW+04], each consisting of 20 genes, with 10 and 11 genes having regulatory interaction, while the remainder moving



(a) The yeast cell cycle subnetwork



(b) Yu's net No. 1 (c) Yu's net No. 2

Figure 1.3: Synthetic Dynamic Bayesian Networks

in a random walk, as depicted in Fig. 1.3(b,c). The data are generated by a simple linear dynamical process as follows:

$$X_{t+1} - X_t = A(X_t - T) + \epsilon \tag{1.4}$$

with X denotes the expression profiles, A describes the strength of gene-gene regulations, T is the constitutive expression values, and ϵ simulates a uniform biological noise. The detailed parameters for each network can be found in [YSW+04]. Using the GeneSim software provided by these authors, we generated, for each number of sample $N = 100, 200$ and 300, 10 data sets for each network. From the average statistics in Table 2.1, it can be seen that Banjo performed well on both data sets. BNFinder achieved a slightly lower sensitivity, but with very high imprecision rate. It is probable that the self-link suppression default option in BNFinder has led the method to include more incorrect edges to the network for a reasonable goodness-of-fit. GlobalMIT performed worse at $N = 100$, but is better at higher number of samples. Again, due to time limit, we were only able to run BNFinder+BDe on one out of ten data sets for each network at $N = 200$ and 300.

1.4.3 Results on Non-Linear Dynamical System Synthetic Data

We employed a five-gene network as in [SI04], of which dynamics is modeled by a system of coupled differential equations adhering to the power-law formalism, called the

S-system. The concrete form of an S-system is given as follows:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}, \quad i = 1 \ldots n, \tag{1.5}$$

where the rates α_i, β_i and kinetic orders g_{ij} and h_{ij} are parameters dictating the influence of gene X_j on the rate of change in the expression level of gene X_i. Using the same system parameters as in [SI04], we integrated the system using the Runge-Kutta method with 10 different initial conditions to obtain 10 time series, each of length 50. We then randomly chose 3 time series of length 33, 3 of length 50 and 6 of length 50, to make data sets of length $N = 99, 150$ and 300 respectively, with 10 data sets for each N value.

Although this data had been previously analyzed with good accuracy by using reconstruction methods based on differential equation models, it proved to be the most challenging case for DBN based methods. Even with a fairly large number of samples, compared to a small number of variables and interactions, all the methods performed poorly, with low sensitivity and high imprecision, rendering the results hardly useful. GlobalMIT nevertheless showed a slight advantage, with a reasonable sensitivity and imprecision at $N = 300$.

1.5 CONCLUSION

This chapter has investigated the problem of learning the globally optimal DBN structure with the MIT scoring metric. We have showed that this task can be achieved using a polynomial time algorithm. Compared with the other well-known scoring metrics, namely BIC/MDL and BDe, both in terms of the worst-case complexity bound and practical evaluation, the BIC/MDL-based algorithm for learning the globally optimal DBN is fastest, followed by MIT, whereas the extensive runtime required by the BDe-based algorithm renders it a very expensive option. GlobalMIT, which is based on a sound information theoretic criterion, represents a very competitive alternative, both in terms of the network quality and runtime required. In the next chapter of this report, we present the application of GlobalMIT on reconstructing the network of biological processes of cyanobacteria.

Table 1.1: Experimental Results

N	GlobalMIT			Banjo			BNFinder+MDL			BNFinder+BDe		
	Sen	Imp	Time	Sen	Imp	Time	Sen	Imp	Time	Sen	Imp	Time
Probabilistic Network Synthetic Data												
30	95 ± 9	29 ± 13	13 ± 3	84 ± 6	70 ± 4	600	86 ± 10	10 ± 9	< 2	85 ± 8	11 ± 11	52 ± 4
70	100 ± 0	1 ± 3	67 ± 4	82 ± 0	51 ± 6	600	100 ± 0	5 ± 7	25 ± 1	100 ± 0	3 ± 4	2.7 ± 0.5h
100	100 ± 0	0 ± 0	499 ± 56	82 ± 0	43 ± 2	600	100 ± 0	1 ± 3	34 ± 1	100	0	9.4h*
Linear Dynamical System Synthetic Data: Yu's net No. 1												
100	54 ± 12	54 ± 13	66 ± 5	58 ± 9	35 ± 16	600	58 ± 9	72 ± 4	4 ± 1	67 ± 7	74 ± 4	4.4 ± 1.3h
200	77 ± 4	19 ± 9	409 ± 127	67 ± 5	8 ± 9	600	66 ± 4	74 ± 2	47 ± 5	67	84	13.6h*
300	79 ± 4	19 ± 12	$.6 \pm .07$h	69 ± 7	4 ± 6	0.6h	68 ± 4	77 ± 2	49 ± 5	67	84	26.5h*
Linear Dynamical System Synthetic Data: Yu's net No. 2												
100	22 ± 15	72 ± 17	44 ± 8	38 ± 11	59 ± 13	600	28 ± 12	83 ± 7	3 ± 1	30 ± 16	86 ± 7	3.3 ± 0.8h
200	49 ± 15	35 ± 19	534 ± 158	45 ± 14	37 ± 16	600	38 ± 8	79 ± 4	39 ± 5	42	85	12.1h*
300	62 ± 12	24 ± 11	$.49 \pm .05$h	53 ± 9	17 ± 13	0.49h	47 ± 9	78 ± 4	40 ± 6	50	85	21.2h*
Non-Linear Dynamical System Synthetic Data												
99	37 ± 10	59 ± 12	< 1	7 ± 3	13 ± 32	600	13 ± 11	81 ± 14	< 1	16 ± 13	77 ± 17	< 1
150	39 ± 16	58 ± 16	< 1	9 ± 11	16 ± 35	600	19 ± 15	71 ± 23	< 1	24 ± 18	67 ± 24	< 1
300	61 ± 7	51 ± 6	< 1	10 ± 12	30 ± 48	600	24 ± 14	74 ± 14	< 1	23 ± 20	80 ± 15	< 1

Sen: percent sensitivity; Imp: percent imprecision; Time: in seconds, unless otherwise specified

*: only run on one data set.

Two

Network Modeling of Cyanobacterial Biological Processes via Clustering and Dynamic Bayesian Network

Cyanobacteria are photosynthetic organisms that are credited with both the initial creation and continuing renewal of the oxygen-rich atmosphere, and are also responsible for more than half of the primary production on earth. Despite their crucial evolutionary and environmental roles, the study of these organisms has lagged behind that of other model organisms. This report presents our ongoing research in unraveling the cyanobacterial network of biological processes. We develop an analysis framework that leverages recently developed bioinformatics and machine learning tools, such as genome-wide sequence matching based annotation, gene ontology analysis, cluster analysis and dynamic Bayesian network. Together, these tools allow us to overcome the lack of knowledge of less well-studied organisms, and reveal interesting relationships among their biological processes. Experiments on the Cyanothece *bacterium demonstrate the practicability and usefulness of our approach.*

2.1 Introduction

Cyanobacteria are the only prokaryotes that are capable of photosynthesis, and are credited with transforming the anaerobic atmosphere to the oxygen-rich atmosphere. They are also responsible for more than half of the total primary production on earth and found the base of the ocean food web. In recent years, cyanobacteria have received increasing interest, due to their efficiency in carbon sequestration and potential for biofuel production. Although their mechanism of photosynthesis is similar to that of higher plants, cyanobacteria are much more efficient as solar energy converters and CO_2 absorbers, essentially due to their simple cellular structure. It is estimated that cyanobacteria are capable of producing 30 times the amount oil per unit area of land, compared to terrestrial oilseed crops such as corn or palm[Oil11]. These organisms therefore may hold the key to solve two of the most fundamental problems of our time, namely climate change and the dwindling fossil fuel reserves.

Despite their evolutionary and environmental importance, the study of cyanobacteria using modern high throughput tools and computational techniques has somewhat lagged behind other model organisms, such as yeast or *E. coli* [SEC+10]. This is reflected partly by the fact that none of the cyanobacteria has an official, effective gene annotation in the Gene Ontology Consortium repository as of May 2011 [The11]. Nearly half of *Synechocystis* sp. PCC 6803's genes, the best studied cyanobacterium, remain unannotated. Of the

annotated genes, the lack of an official, systematic annotating mechanism, such as that currently practiced by the Gene Ontology Consortium, make it hard to verify the credibility of the annotation as well as to perform certain type of analysis, e.g., excluding a certain annotation evidence code.

In this work, to alleviate the difficulties faced when studying novel, less well-studied organisms such as cyanobacteria, we develop an analysis framework for building network of biological processes from gene expression data, that leverages several recently developed bioinformatics and machine learning tools. The approach is divided into three stages:

- Filtering and clustering of genes into clusters which have coherent expression pattern profiles. For this, we propose using an automated scheme for determining a suitable number of clusters for the next stages of analysis.
- Assessment of clustering results using functional enrichment analysis based on gene ontology. Herein, we propose using annotation data obtained from two different sources: one from the Cyanobase cyanobacteria database [Kaz11], and another obtained by means of computational analysis, specifically by amino sequence matching, as provided by the Blast2GO software suite [GGGT+08].
- Building a network of interacting clusters. This is done using the recently developed formalism of dynamic Bayesian network (DBN). We apply our recently proposed GlobalMIT algorithm for learning the globally optimal DBN structure from time series microarray data, using an information theoretic based scoring metric.

It is expected that the network of interacting clusters will reveal the interactions between biological processes represented by these clusters. However, when doing analysis on the cluster (or biological process) level, we lose information on individual genes. Obtaining such information is possible if we apply network reverse engineering algorithms directly to the original set of genes without clustering, resulting in the underlying gene regulatory network (GRN). Nevertheless, with a large number of genes and a limited number of experiments as often seen in microarray data, GRN-learning algorithms face severe difficulties in correctly recovering the underlying network. Also, a large number of genes (including lots of unannotated genes) makes the interpretation of the results a difficult task. Analysis at the cluster level serves two purposes: (i) to reduce the number of variables, thus making the network learning task more accurate, (ii) to facilitate interpretation. Similar strategies to this approach have also been employed in [SSR+03, dHIK+03, SEC+10].

In the rest of this report, we present our detailed approach for filtering and clustering of genes, assessing clustering results, and finally building network of interacting clusters. For an experimental cyanobacterial organism, we chose the diazotrophic unicellular *Cyanothece* sp. strain ATCC 51142, hereafter *Cyanothece*. This cyanobacterium represents a relatively less well-studied organism, but with a very interesting capability of performing both nitrogen fixation and photosynthesis within a single cell, the two processes that are at odds with each other [SWL+08].

2.2 Filtering and clustering of genes

Cyanobacteria microarray data often contain measurements for 3000 to 6000 genes. Many of these genes, such as house keeping genes, are not expressed, or expressed at a constant level throughout the experiments. For analysis, it is desirable to filter out these genes, and retain only genes which are differentially expressed. There are various methods for

filtering genes such as the threshold filter, Student's t-test or analysis of variance (ANOVA) [EPW+10]. In this work, we implement a simple but widely employed threshold filter to remove genes that are not differentially expressed above a certain threshold throughout the experimental process, e.g., 1.5-fold or 2-fold change.

Next, we cluster the selected genes into groups of similar pattern profiles. In the recent years, there has been dozens of clustering algorithms specifically developed for the purpose of clustering microarray data. Some of the most popular methods include K-means, hierarchical clustering, self organizing map, graph theoretic based approaches (spectral clustering, CLICK, CAST), model based clustering (mixture models), density based approaches (DENCLUE) and affinity propagation based approaches [JTZ04]. In this work, we implement the widely used K-means with log-transformed microarray data.

A crucial parameter for K-means type algorithms is the number of clusters K. For our purpose in this work, K will control the level of granularity of the next stages of analysis. We use our recently developed Consensus Index for automatically determining the relevant number of clusters from the data [VEB10]. The Consensus Index (CI) is a realization of a class of methods for model selection by stability assessment [ST08], whose main idea can be summarized as follows: for each value of K, we generate a set of clustering solutions, either by using different initial starting points for K-means, or by a certain perturbation scheme such as sub-sampling or projection. In regard to the set of clusterings obtained, when the specified number of clusters coincides with the "true" number of clusters, this set has a tendency to be less diverse—an indication of the robustness of the obtained cluster structure. The Consensus Index was developed to quantify this diversity. Specifically, given a value of K, suppose we have generated a set of B clustering solutions $\mathcal{U}_K = \{U_1, U_2, ..., U_B\}$, each with K clusters. The consensus index of $\mathcal{U}_K$ is defined as:

$$\mathrm{CI}(\mathcal{U}_K) = \frac{\sum_{i<j} \mathrm{AM}(\mathbf{U}_i, \mathbf{U}_j)}{B(B-1)/2} \tag{2.1}$$

where the agreement measure AM is a clustering similarity measure. In this work, we use the Adjusted Rand Index (ARI) and the Adjusted Mutual Information (AMI—which is the adjusted-for-chance version of the widely used Normalized Mutual Information) as clustering similarity measures [VEB09]. The optimal number of clusters K^* is chosen as the one that maximizes CI, i.e., $K^* = \arg\max_{K=2...K_{max}} \mathrm{CI}(\mathcal{U}_K)$ where K_{max} is the maximum number of clusters to be considered.

2.3 ASSESSMENT OF CLUSTERING RESULTS

Having obtained a reasonable clustering solution, we next investigate the major biological functions of each cluster. In this work, this is done by means of functional enrichment analysis using gene ontology (GO), where every GO terms appearing in each cluster is assessed to find out whether a certain functional category is significantly over-represented in a certain cluster, more than what would be expected by chance. To do this, first of all, we need a genome-wide annotation of genes in the organism of interest. As stated previously, one of the difficulties working with less well-studied organisms is that there is not an official annotation database. To address this challenge, we propose gathering annotation data from two different sources: one from the Cyanobase database [Kaz11], and another from genome-wide amino sequence matching using the Blast2GO software suit [GGGT+08]. We describe each source below.

The Cyanobase maintains, for each cyanobacterium in its database, an annotation file which was obtained by IPR2GO, a manually-curated mapping of InterPro terms to GO terms that is maintained by the InterPro consortium [The02]. Although being the result of a manual curation process, surprisingly, it has been reported that the accuracy of this mapping can be considerably lower than some automated algorithms, such as that reported in [JT06]. Moreover, the number of annotated genes normally accounts for just less than half of the genome, eg. in the case of *Cyanothece*, there are currently only annotations for 2566 genes out of 5359 genes (as of May 2011).

Thus, in order to supplement the Cyanobase IPR2GO annotation, we employ Blast2GO, a software suit for automated gene annotation based on sequence matching [GGGT+08]. Blast2GO uses BLAST search to find similar sequences to the sequence of interest. It then extracts the GO terms associated to each of the obtained hits and return the GO annotation for the query. For *Cyanothece*, Blast2GO was able to supplement the annotation for almost another one thousand genes. In this work, we aggregate Cyanobase IPR2GO and Blast2GO annotation into a single pool, then use BiNGO [MHK] for GO functional category enrichment analysis. For BiNGO, we use the filtered gene set as the reference set, the hypergeometric test as the test for functional over-representation, and False Discovery Rate (FDR) as the multiple hypothesis testing correction scheme.

2.4 Building a network of interacting clusters

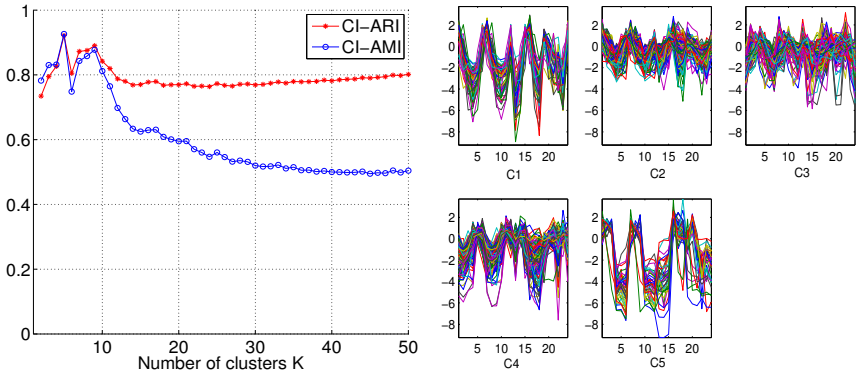
Our next step is to build a network of interacting clusters, in order to understand the connections between biological processes represented by these clusters. We perform this modeling task using the recently developed dynamic Bayesian network (DBN) formalism [MM99, Hus03, YSW+04, ZC05]. DBN addresses two weaknesses of the traditional static Bayesian network (BN) model in that (i) it allows feedback loops, which are prevalent in biological systems, and (ii) has an explicit mechanism for exploiting time-series microarray data. We employ our GlobalMIT algorithm developed in the previous chapter for this task.

2.5 Experiments on *Cyanothece* sp. strain ATCC 51142

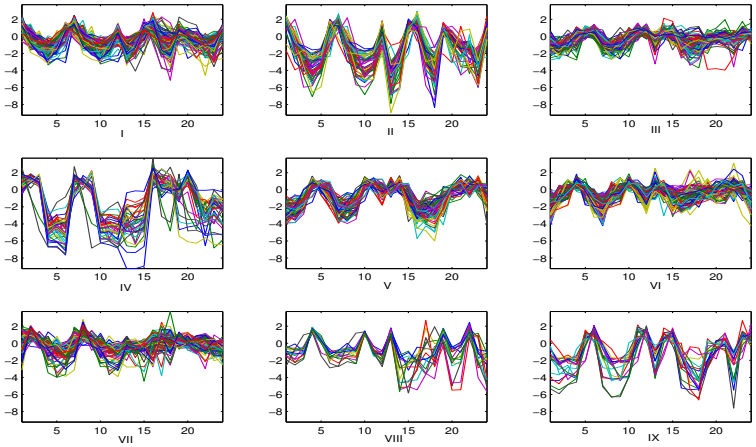
In this section, we present our experimental results on *Cyanothece*. We collected two publicly available genome-wide microarray data sets of *Cyanothece*, performed in alternating light-dark (LD) cycles with samples collected every 4h over a 48h period: the first one starting with 1h into dark period followed by two DL cycles (DLDL), and the second one starting with two hours into light period, followed by one LD and one continuous LL cycle (LDLL) [WGP11]. In total, there were 24 experiments.

Filtering and clustering of genes: Using a threshold filter with a 2-fold change cutoff, we selected 730 genes for further analysis. We first used the Consensus Index to determine the number of clusters in this set. Fig. 2.1(a) show the CI with $K \in [2, 50]$. It can be seen that the CI with both the ARI and AMI strongly suggests $K = 5$ (corresponding to the global peak). Also, a local peak is present at $K = 9$. As discussed in [VEB10], the local peak may correspond be the result of the hierarchical clustering structure in the data. We performed K-means clustering with both $K = 5$ and $K = 9$, each for 1 million times with random initialization, and picked the best clustering results, presented in Fig. 2.1(b,c).

Assessment of clustering results: From the visual representation in Fig. 2.1(b), it can be seen that the clusters have distinct pattern profiles. GO analysis of the clustering results



(a) Consensus Index (b) Clustering result with $K = 5$



(c) Clustering result with $K = 9$

Figure 2.1: Cluster analysis of *Cyanothece* microarray data

are presented in Tables 2.1 and 2.2. From Table 2.1, of our particular interest is cluster C5, which is relatively small but contains genes exclusively involved in the nitrogen fixation process. It is known that *Cyanothece* sp. strain ATCC 51142 is among the few organisms that are capable of performing both oxygenic photosynthesis and nitrogen fixation in the same cell. Since the nitrogenase enzyme involved in N_2 fixation is inactivated when exposed to oxygen, *Cyanothece* separates these processes temporally, so that oxygenic photosynthesis occurs during the day, and nitrogen fixation during the night. Cluster C4 is also of our interest, since its contains a large number of genes involved in photosynthesis. As the experimental condition involves alternative light-dark condition, it could be expected that the genes involved in nitrogen fixation and photosynthesis will strongly regulate each other, in the sense that the up-regulation of N_2 fixation genes will lead to the down-regulation of photosynthesis genes, and vice-versa.

Building a network of interacting clusters: We apply the GlobalMIT algorithm to learn

Table 2.1: GO analysis of the 5-cluster clustering results

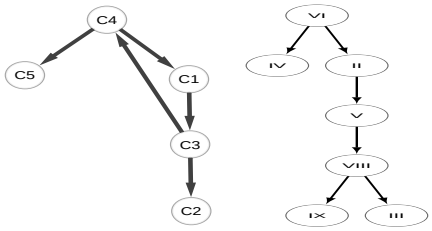
Cluster	Size	GO ID	Description	#Genes	Corrected P-value
C1	54	8746	NAD(P)+transhydrogenase activity	3	0.98%
		70469	respiratory chain	3	2.9%
C2	206	8652	cellular amino acid biosynthesis	18	2.1%
		4518	nuclease activity	8	4.1%
C3	236	32991	macromolecule complex	61	2E-10
		30529	ribonucleoprotein complex	24	2.9E-7
		6412	translation	29	1.5E-6
		44267	cellular protein metabolic process	46	5.4E-5
		19538	protein metabolic process	50	0.14%
C4	196	71944	cell periphery	35	6.8E-5
		9512	photosystem	20	6.5E-3
		6022	aminoglycan metabolic process	6	2%
C5	38	9399	nitrogen fixation	19	5.7E-22
		51536	iron-sulfur cluster binding	12	9.6E-6
		16163	nitrogenase activity	5	1.5E-5

Table 2.2: GO analysis of the 9-cluster clustering results

Cluster	Size	GO ID	Description	#Genes	Corrected P-value
I	157	8652	cellular amino acid biosynthesis	17	4.5E-3
		46394	carboxylic acid biosynthesis	17	1.6%
II	48	55114	oxidation reduction process	14	17%
		15980	energy derivation by oxidation	6	17%
III	127	15979	photosynthesis	17	45%
		6810	transport	27	45%
IV	36	9399	nitrogen fixation	19	1.5E-24
V	68	6022	aminoglycan metabolic process	5	2.1%
		7049	cell cycle	4	3.8%
VI	158	15979	photosynthesis	36	3.6E-10
VII	101	6412	translation	27	6.7E-13
VIII	16	65007	biological regulation	5	7.7%
		51171	regulation of nitrogen compound	3	7.7%
IX	19	15706	nitrate transport	3	2.2E-3
		6810	transport	9	2.2%

a DBN structure, first to the 5-cluster clustering result. We take the mean expression value of each cluster as its representative. There are thus 5 variables over 24 time-points fed into GlobalMIT. The globally optimal DBN network as found by GlobalMIT is presented on Fig. 2.2(a). It is readily verifiable the fact that nitrogen fixation genes and photosynthesis genes strongly regulate each other, since there is a link between cluster C4 (photosystem) and C5 (nitrogen fixation).

We perform a similar analysis on the 9-cluster clustering result. The DBN for this 9-cluster set is presented in Fig. 2.2(a). Note that clusters I and VII are disconnected. We are interested in verifying whether the link between the photosynthesis cluster and the nitrogen fixation cluster remains at this level of granularity. Visually, it it easily recognizable from Fig. 2.1(b-c) that cluster C5 in the 5-cluster set corresponds to cluster IV in the 9-cluster set. GO analysis on cluster IV confirms this observation (Table 2.2). We therefore pay special attention to clusters VI, since there are a link between cluster VI and IV. Not surprisingly, GO analysis reveal that cluster VI contains a large number of genes involved in photosynthesis. The structural similarity between the two graphs is also evident from Fig. 2.2. At a higher level of granularity, the clusters become more specialized. The links between cluster VIII and {IX, III} are also of interest, since cluster VIII is a tightly co-regulated group which contains several genes with regulation activities, which might regulate genes involving transport and photosynthesis (clusters III and IX).



(a) 5-cluster set (b) 9-cluster set

Figure 2.2: DBN analysis of *Cyanothece* clustered data

2.6 Discussion and Conclusion

In this report, we have presented an analysis framework for unraveling the interactions between biological processes of novel, less well-studied organisms such as cyanobacteria. The framework harnesses several recently developed bioinformatics and data mining tools to overcome the lack of information of these organisms. Via Blast2GO and IPR2GO, we could collect annotation information for a large number of genes. Cluster analysis helps to bring down the number of variables for the subsequent network analysis phase, and also facilitates interpretation. We have demonstrated the applicability of our framework on *cyanothece*. Our future work involves further analysis of other cyanobacteria that are potential for carbon sequestration and biofuel production.

Acknowledgments

This project is supported by an Australia-India strategic research fund (AISRF).

Availability

Our implementation of the algorithms proposed in this report in Matlab and C++ is available at http://code.google.com/p/globalmit.

Bibliography

[Chi96] David M. Chickering. Learning Bayesian Networks is NP-Complete. In D. Fisher and H. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. 1996.

[dC06] Luis M. de Campos. A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *J. Mach. Learn. Res.*, 7:2149–2187, December 2006.

[dHIK+03] M.J.L. de Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano. Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. *Pac Symp Biocomput*, pages 17 – 28, 2003.

[DLH10] Frank Dondelinger, Sophie Lebre, and Dirk Husmeier. Heterogeneous continuous dynamic bayesian networks with flexible structure and inter-time segment information sharing. In *ICML*, pages 303–310, 2010.

[Doj06] Norbert Dojer. Learning Bayesian Networks Does Not Have to Be NP-Hard. In *Proceedings of International Symposium on Mathematical Foundations of Computer Science*, pages 305–314, 2006.

[EPW+10] T.R. Elvitigala, A.D. Polpitiya, Wenxue Wang, J. Sto? andckel, A. Khandelwal, R.S. Quatrano, H.B. Pakrasi, and B.K. Ghosh. High-throughput biological data analysis. *Control Systems, IEEE*, 30(6):81 –100, dec. 2010.

[GGGT+08] Stefan Gotz, Juan Miguel Garcia-Gomez, Javier Terol, Tim D. Williams, Shivashankar H. Nagaraj, Maria Jose Nueda, Montserrat Robles, Manuel Talon, Joaquin Dopazo, and Ana Conesa. High-throughput functional annotation and data mining with the Blast2GO suite. *Nucleic Acids Research*, 36(10):3420–3435, 2008.

[GH09] Marco Grzegorczyk and Dirk Husmeier. Non-stationary continuous dynamic Bayesian networks. In *NIPS 2009*, 2009.

[Har] Alexander Hartemink. Banjo: A structure learner for static and dynamic bayesian networks.

[Hus03] Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19(17):2271–2282, 2003.

[JT06] Jaehee Jung and Michael Thon. Automatic annotation of protein functional class from sparse and imbalanced data sets. In Mehmet Dalkilic, Sun Kim, and Jiong Yang, editors, *Data Mining and Bioinformatics*, volume 4316 of *Lecture Notes in Computer Science*, pages 65–77. Springer Berlin / Heidelberg, 2006.

[JTZ04] Daxin Jiang, Chun Tang, and Aidong Zhang. Cluster analysis for gene expression data: a survey. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1370 – 1386, nov. 2004.

[Kaz11] Kazusa DNA Research Institute. The cyanobacteria database. 2011.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[KIM03] Sun Yong Kim, Seiya Imoto, and Satoru Miyano. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Briefings in Bioinformatics*, 4(3):228–235, 2003.

[Kul68] Solomon Kullback. *Information Theory and Statistics*. Dover publications, 1968.

[MHK] Steven Maere, Karel Heymans, and Martin Kuiper. BiNGO: a cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21(16):3448–3449.

[MM99] Kevin Murphy and Saira Mian. Modelling gene expression data using dynamic bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA, 1999.

[Oil11] Oilgea Inc. Comprehensive oilgae report. 2011.

[PRM+03] Bruno-Edouard Perrin, Liva Ralaivola, AurÃľlien Mazurie, Samuele Bottani, Jacques Mallet, and Florence dâĂŹAlchÃľâĂŞBuc. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19(suppl 2):ii138–ii148, 2003.

[RH09] Joshua Robinson and Alexander Hartemink. Non-stationary dynamic Bayesian networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 1369–76, 2009.

[RH10] Joshua Robinson and Alexander Hartemink. Learning Non-Stationary Dynamic Bayesian Networks. In *the Journal of Machine Learning Research*, volume 11, pages 3647–3680, 2010.

[SEC+10] Abhay Singh, Thanura Elvitigala, Jeffrey Cameron, Bijoy Ghosh, Maitrayee Bhattacharyya-Pakrasi, and Himadri Pakrasi. Integrative analysis of large scale expression profiles reveals core transcriptional response and coordination between multiple cellular processes in a cyanobacterium. *BMC Systems Biology*, 4(1):105, 2010.

[SI04] Naoya Sugimoto and Hitoshi Iba. Inference of gene regulatory networks by means of dynamic differential bayesian networks and nonparametric regression. *Genome Informatics*, 15(2):121–30, 2004.

[SSR+03] Eran Segal, Michael Shapira, Aviv Regev, Dana Pe'er, David Botstein, Daphne Koller, and Nir Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34:166 – 176, 2003.

[ST08] Ohad Shamir and Naftali Tishby. Model selection and stability in k-means clustering. In *COLT'08*. Springer, 2008.

[SWL+08] Jana Stockel, Eric A. Welsh, Michelle Liberton, Rangesh Kunnvakkam, Rajeev Aurora, and Himadri B. Pakrasi. Global transcriptomic analysis of cyanothece 51142 reveals robust diurnal oscillation of central metabolic processes. *Proceedings of the National Academy of Sciences*, 105(16):6156–6161, 2008.

[The02] The InterPro Consortium. Interpro: An integrated documentation resource for protein families, domains and functional sites. *Briefings in Bioinformatics*, 3(3):225–235, 2002.

[The11] The Gene Ontology Consortium. Current annotations. 2011.

[VEB09] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1073–1080, New York, NY, USA, 2009. ACM.

[VEB10] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.

[WD09] Bartek Wilczynski and Norbert Dojer. BNFinder: exact and efficient method for learning Bayesian networks. *Bioinformatics*, 25(2):286–287, 2009.

[WGP11] Wenxue Wang, B.K. Ghosh, and H. Pakrasi. Identification and modeling of genes with diurnal oscillations from microarray time series data. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8(1):108 –121, jan.-feb. 2011.

[YSW+04] Jing Yu, V. Anne Smith, Paul P. Wang, Alexander J. Hartemink, and Erich D. Jarvis. Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–3603, 2004.

[ZC05] Min Zou and Suzanne D. Conzen. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21(1):71–79, 2005.