

# A Framework to Learn Bayesian Network from Changing, Multiple-Source Biomedical Data

Guoliang Li and Tze-Yun Leong

Medical Computing Laboratory, School of Computing, National University of Singapore  
3 Science Drive 2, Singapore, 117543  
{ligl, leongty}@comp.nus.edu.sg

## Abstract

Structure learning in Bayesian network is a big issue. Many efforts have tried to solve this problem and quite a few algorithms have been proposed. However, when we attempt to apply the existing methods to microarray data, there are three main challenges: 1) there are many variables in the data set, 2) the sample size is small, and 3) microarray data are changing from experiment to experiment and new data are available quickly. To address these three problems, we assume that the major functions of a kind of cells do not change too much in different experiments, and propose a framework to learn Bayesian network from data with variable grouping. This framework has several advantages: 1) it reduces the number of variables and narrows down the search space when learning Bayesian network structure; 2) it relieves the requirement for the number of samples; and 3) the learned group Bayesian network is a higher-level abstraction of biological functions in a cell, which is comparable from one experiment to another, and does not need to change much at the level when the learned group Bayesian network is applied to changing experiments - only the relationship between a group variable and an original variable should be adjusted. We have done experiments on synthetic examples and real data to test the proposed framework. The preliminary results from synthetic examples show that the framework works with fewer samples, and the learned group Bayesian networks from different sets of experimental data agree with each other most of the time. The experiments with the real data also show some domain-meaningful results. This framework can also be applied to other domains with similar assumptions.

## Introduction

Bayesian network is a graphical tool to model uncertainty and infer causal relationship among variables. Usually it is constructed with domain knowledge. In recent years, especially for situations where domain knowledge is insufficient, many efforts have tried to learn Bayesian network from data (Chickering & Heckerman, 1997; Cooper & Herskovits, 1992; Dash & Druzdzel, 1999; Friedman, 1998; Lu, Druzdzel, & Leong, 2000; Neapolitan, 2004; Segal, Pe'er, Regev, Koller, &

Friedman, 2003) with tools for constructing and learning Bayesian network. Refer to (Murphy, 2004) for summary. However, the structure learning algorithms in these tools only take tens (or a little more) of variables into consideration. They are inadequate for domains which require hypotheses of causal relationships among hundreds (even thousands) of variables, such as microarray data analysis.

The problem in microarray data analysis is to infer the possible relationships among genes or gene groups. There are quite a few methods proposed for microarray data analysis, such as statistical methods, clustering methods. Among the proposed methods, Bayesian network is a promising one (Friedman et al, 2000; Joshi, Leong, 2005), since Bayesian network is easy to interpret and could possibly support discovery of causal relationships among the genes, and the relationships among genes are stochastic from biological knowledge. However, when we want to apply the existing Bayesian networks learning methods to microarray data, there are three main challenges: 1) there are many variables in the data set, 2) the sample size is small and 3) microarray data are changing from experiment to experiment, and new data are available quickly. These challenges are not uncommon in the Bayesian network domain, but the third challenge is of special significance here. Different biological research groups do microarray experiments for different purposes and new microarray data are emerging quickly. Since the conditions in these experiments are quite different, it is not meaningful to combine these data sets directly into one big data set. Moreover, the Bayesian networks learned from different data sets are not directly comparable. To maximally utilize the interpretability and causal-relationship-discovery capabilities of Bayesian network, we need to extend the Bayesian network formalism for microarray data analysis.

From biological knowledge, we know that genes in a cell can be partitioned into different groups, and each group of genes performs a particular biological function. In our work, we take this knowledge into consideration and assume that the major functions of certain cells do not change too much in different experimental conditions. Then, we introduce group variables to represent different groups of genes and propose a framework to learn a

Bayesian network to represent the relationships among different groups. The values of a group variable are the activity level of the corresponding biological function performed by this group of genes, which will be learned from the original data. A Bayesian network will be learned with the group variables only. We call the learned Bayesian network a group Bayesian network. This framework has several advantages. First, it will reduce the number of variables when only group variables are used to learn the Bayesian network structure. Reducing the number of variables can narrow down the structure space of Bayesian network. Second, it will relieve the requirement for the number of samples. Moreover, the learned group Bayesian network is a high-level abstraction of biological functions in a cell. The group variables are more reliable to represent the functions in a cell and the Bayesian network learned with group variables will be more consistent in different experiment data sets.

In this paper, we have conducted experiments on synthetic examples and real data. The results from synthetic examples show that the framework can work with samples in median sample size and identify the expected group Bayesian network from different data sets – The expected group Bayesian network has the highest BIC score. The experiments from the real microarray data show some domain-meaningful results. This framework can also be applied to other domains with similar assumptions.

## Literature Review

Bayesian network structure learning is a big topic in the area of uncertainty in artificial intelligence. Learning Bayesian network from data has been studied for more than ten years and many methods have been proposed. The problem can be divided into two categories based on the properties of the data. When the data is complete, the structure learning is a discrete optimization problem over structures. Discrete search is the common method for this problem. When the data is incomplete, it is necessary to estimate the missing values while learning the Bayesian network structure. Structural EM, mixture models and approximate methods have been proposed for this type of problems (Cheeseman & Stutz, 1996; Chickering & Heckerman, 1997; Friedman, 1998).

To learn a Bayesian network structure from data, there are two different approaches in general: search-and-score-based approach and constraint-based approach. In search-and-score-based approach, the algorithms search over the possible structure space, score each structure with the given data, and give as a result the structure with the best score or the distribution of the scored structures.

The followings are some representative search-and-score-based methods. The simple one is exhaustive search. Since the number of possible Bayesian network structure is exponential to the number of the variables, we cannot do exhaustive search with a reasonable size of variables. The second algorithm is the K2 algorithm (Cooper &

Herskovits, 1992). It is shown to be theoretically sound, with some assumptions, and can learn the structure quickly. However, the assumption that the ordering of the variables is known beforehand is very strong, and cannot be satisfied in many conditions. The third one is Greedy Search (Chickering, 2002). It starts from an initial structure and moves to the direct neighbors with the highest score that is better than the current one, until it reaches a local maximum. Markov Chain Monte Carlo (MCMC) (Madigan & York, 1995) samples over the structure space, and gives a posterior distribution of the scored structures. Structural Estimation-Maximization (EM) (Friedman, 1998) searches over the structure and parameter simultaneously. There are some other methods available in the literature that are not covered here.

In the constraint-based approach, the algorithms perform many conditional independence tests among variables and infer the Bayesian network structure with these conditional independences. The representative algorithms are the Inductive Causation (IC) algorithm (Pearl & Verma, 1991), Spirtes-Glymour-Scheines (SGS) algorithm, PC algorithm (Spirtes, Glymour, & Scheines, 1993), and etc.

## Method

When we attempt to apply the existing Bayesian network structure learning methods to microarray data, there are some problems. First, there are many variables in microarray data – more than that the existing methods can deal with. Second, there is an assumption in these existing Bayesian network learning methods: the number of samples is 10 times (or more) of the number of variables in the data set. However, in microarray data sets, the number of samples does not satisfy this requirement. Hence the existing Bayesian network structure learning methods can not be applied to microarray data set directly. Moreover, microarray data sets may be from different research groups, prepared under different conditions. Therefore, new methods are required to learn Bayesian network from microarray data sets. In this work, we assume that the data set is complete. If there are missing values, we preprocess the data and fill in the missing values with the means of the corresponding variables' existing values.

## Observations on Microarray Data

Microarray is a technology used in biological experiment, which can simultaneously measure the activity levels of thousands of genes in the cell under a particular condition. The measured results are microarray data, in which each gene is treated as a variable and each experiment is a sample. The data set usually has hundreds or thousands of genes but only hundreds (even tens) of experiments (samples).

To learn a Bayesian network structure from microarray data, we examine the data first and note the following observations from the data and the domain knowledge. First, we know that some genes have similar functions

from biological knowledge, e.g., the genes in the same gene complex. These redundant functions among genes guarantee that the defect of some genes cannot degrade the functions of an entire cell too much. Second, the expression levels of such genes are similar or related under different conditions. Third, some genes act together to perform a biological function. This means that genes can be partitioned into groups according to their functions. Moreover, the genes in a group interact with genes in other groups, and the entire interactions among the groups are more important than those among the individual genes.

## The Framework

Based on the above observations, we propose the following framework to learn Bayesian network with a variable grouping method. The framework includes four steps: 1) Partition the original variables into different groups; 2) Determine the representative value for each group. Group variables are introduced as hidden variables to represent each group in this step; 3) Learn a Bayesian network with group variables only; and 4) Recover the possible structure of the entire variables from the learned group Bayesian network structure. Since the number of the group variables is (much) smaller than the number of the original variables, the Bayesian network learning process will be speeded up, and the requirement for samples will be relieved. Although the idea is motivated by the domain of microarray data analysis, the framework can be applied to other data with the similar situations. The details of each step will be discussed in the following sub-sections.

### Partition the Original Variables into Different Groups.

The first step in the framework is to group the original variables: partition  $n$  variables into  $m$  groups ( $m < n$ ). It can be done according to two different conditions – with or without domain knowledge. If we have enough domain knowledge and know the number of groups and membership of each original variable, we can assign them directly and this step will be completed easily.

If we do not have enough domain knowledge and do not know the possible groups in the original variables, we need to learn the assignment of each variable to a group – which is similar to variable clustering. The difference between the variable grouping in this framework and the ordinary variable clustering is in the evaluation step leading to grouping. The evaluation step in this method is based on the learned Bayesian network from the corresponding group variables, not the ordinary clustering score metrics. We need to adjust the variable grouping based on the criterion for the learned Bayesian network. Certainly, the result from the ordinary variable clustering can be used as the initial point in greedy grouping – which will be discussed later.

#### 1) Exhaustive search for variable grouping

The intuitive way to group variables is to enumerate all the possible configurations to partition  $n$  variables into  $m$  groups. Since the number of the possible configurations is exponential, we cannot do exhaustive search for

moderately large  $n$  and  $m$ . This method is implemented as a golden standard for testing the small cases only.

#### 2) Greedy search for grouping – greedy grouping

Since the grouping space is exponential in the number of variables and the number of groups, we need heuristics to speed up the search. Greedy search for grouping – greedy grouping – is adopted in this work. The process is as below. First, greedy grouping starts from an initial point of the grouping space – a specific assignment of the variables to groups. The initial assignment may be generated randomly, derived from an ordinary clustering method, from domain knowledge or combined with domain knowledge. Second, the algorithm tests all nearest neighbors of the current grouping – all possible group assignment with only one original variable changed from one group to another group in the current grouping. Third, the algorithm chooses as the current grouping the neighbor with the highest score and better than the current grouping. The score is measured on the learned Bayesian network with group variables only. Lastly, the grouping process stops when no neighbors have higher score than the current grouping. In step 2, only one variable's assignment is changed and two groups are involved. The results for other groups can be cached to speed up the process.

Greedy grouping does the optimization locally and always reaches a local maximum – there is no guarantee to reach a global maximum. To escape from the local maximum, we can restart the greedy search several times with other initial points and select the best result we can get.

#### 3) Combine domain knowledge in grouping

There are several different conditions about domain knowledge. If we have the complete knowledge about the grouping of the variables, the grouping is done. If we are not certain about the grouping, we can use the grouping from domain knowledge as the initial assignment in greedy grouping. If we only have partial domain knowledge, such as two variables are definitely in the same group or in two different groups, we can add constraints in greedy grouping to make the grouping result to be consistent with the domain knowledge.

### Determine the representative values for each group.

The representative values of each group are represented as group variables in this work. The group variables are assumed to be hidden variables. Compared with the original variables in the microarray data, the values of the group variables are more reliable in different experiments. Especially, when the microarray data are obtained from different changing environments, the expression levels of different individual genes may be quite different. But the relationships among the groups of genes will still be quite consistent.

Determining the representative values for the group variables is an essential step in the proposed method, since the following Bayesian network structure learning is based on the group variables. Their values can be determined by the following ways. 1) As hidden variable in Bayesian network. One hidden variable is learned for each group

and as the root of the Bayesian network. The variables can be continuous and/or discrete. AutoClass (Cheeseman & Stutz, 1996) has been tried for this purpose. 2) Center of each group. It is assumed that the variables in each group are continuous in this case – as that in microarray data. 3) First principal component of each group. The principal component analysis (PCA) method is used here. It is assumed that the variables are all continuous in this case, too. 4) Select a representative variable from each group; 5) Learned from other learning methods; 6) From domain knowledge (experts or knowledge base). Currently we have implemented the second and third ways in our method.

**Learn a Bayesian network based on the group variables.** The third step is to learn a Bayesian network with the group variables only. In this step, any existing Bayesian network learning algorithms can be used, such as greedy search with restart, evaluated with BIC score. The important issue in this step is that the Bayesian network structure learning is based on group variables only. No original variables are used in this step. We name the learned Bayesian network the group Bayesian network.

**Recover the possible structure of the entire variables from the group Bayesian network.** The fourth step is to recover the structure with all the variables. There are two strategies in this step. One strategy is to keep the group variables in the recovered structure and the other one is to eliminate the group variables from the recovered structure. In the framework, a local structure is defined for possible structure recovery purpose – it is the structure between a group variable and the original variables in the corresponding group. In Step 2 of the proposed method, a local structure between the group variable and the original variables in each group is learned (with PCA, Bayesian network, or other methods) or assumed. When the representative value of the group is from PCA or the center of the original variables of the group, the local structure is a tree structure – the original variables are independent of each other given the group variable. When the representative value is learned from Bayesian network, the original variables may be not independent of each other given the group variable. Currently the variables in one group are independent of each other given the group variable for the simplicity reason in the implementation.

In the first strategy, the local structures are concatenated to the group Bayesian network to form an entire Bayesian network. In the second strategy, if two group variables are connected to each other directly, d-separation nodes between the original variables in these two groups are determined by conditional independence test. The variables in two groups are connected by the d-separation nodes.

In both strategies, the structure among the group variables is the main frame of the recovered Bayesian network. If some groups contain only one original variable, the group variables are substituted with the corresponding original variables directly.

## Important Issues in the Proposed Method

There are two important issues to be mentioned here. First, there are two search spaces in our method – one for grouping the variables, and the other for the possible Bayesian network structures of the group variables. Although both of these two spaces are exponential (one in the number of original variables and the number of the groups, and the other in the number of group variables), the combination space is much narrower than the space of the possible Bayesian network structures with the entire original variables. Second, several heuristics are used in the process. Aside from the heuristics in the existing Bayesian network structure learning methods, we adopt greedy search for variable grouping and cache the unchanged groups in greedy grouping. These heuristics make the process reach a local maximum faster.

## Theoretical Correctness

In the learning process, we apply greedy search for variable grouping and structure learning for Group Bayesian network. We always choose a grouping and group Bayesian network with higher score as the next group assignment and structure. The score never decreases. When the algorithm stops, it is guaranteed to reach a local maximum.

## Experiment

The proposed method has been tested in experiments with synthetic examples and real data. In the synthetic examples, we build an artificial model. Then we sample data from the model, learn a group Bayesian network from the sampled data, and compare the learned model with the expected model.

**Synthetic Example** Figure 1(a) shows a small example. There are 4 Gaussian variables in this model. Variable 1 follows a normal distribution with 0 mean and unit standard deviation –  $\text{var}1 \sim N(0,1)$ . The means of variable 2 and variable 3 are dependent on the value of variable 1 –  $\text{var}2 \sim N(\text{var}1, 1)$  and  $\text{var}3 \sim N(\text{var}1, 1)$ . The mean of variable 4 is dependent on the sum of the values of the variable 2 and variable 3 –  $\text{var}4 \sim N(\text{var}2 + \text{var}3, 1)$ . Here variable 2 and 3 follow the same conditional probability distribution, and are similar to each other. So they should be grouped together in the group Bayesian network. We sampled 11 cases from the model to learn a group Bayesian network. The number of samples is smaller than the requirement for the existing Bayesian network structure learning methods (10 times or more than the number of variables).

In the experiment, exhaustive search over grouping was tested first. The grouping problem here is to partition 4 variables into 3 different groups. There are 6 different cases in total. Figure 1(b) shows the structure with the highest BIC score. Group 1 contains variable 1, group 2 contains variables 2 and 3, and group 3 contains variable 4.

This is the same as what we expect. Greedy search over grouping has also been tested on this example. The result is the same as the result from exhaustive search over grouping. We repeated the experiments 221 times: sampling 11 cases from the synthetic model and learning a group Bayesian network from the samples. This process is like doing the different experiments to collect microarray data. The results show that in 82.8% of the cases, the learned grouping is the same as expected and the extended Bayesian network is shown in Figure 1(b).

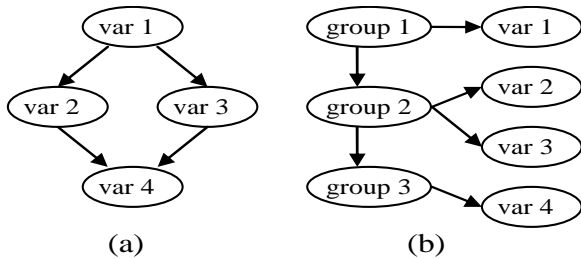


Figure 1 A simple synthetic example

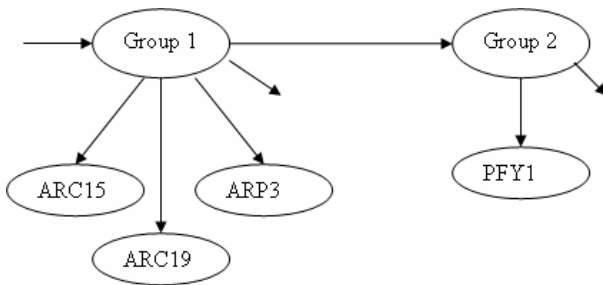


Figure 2 A partial graph from the learned model. Group 1 and 2 contain more genes than those in the figure. The arrows without nodes at the beginning or end mean that some nodes are not displayed here.

**Microarray Data** The microarray data set used in this work is that used in (Gasch et al., 2000), which measured the response of yeast cells to environmental changes under different conditions. The data set contains 6157 genes and 173 experiments. From this data set, we selected 90 known genes in Actin cytoskeleton group to learn a group Bayesian network for testing purpose. The missing values in the data set were filled in with the mean of the known values for each gene. The number of groups is set to 15 in our experiment. We ran the experiments ten times to test whether the learned groups and group Bayesian networks are consistent in the majority of the experiments. For example, ARC15, ARC19, ARP3 and the other 3 genes are in one of the learned groups up all the experiments. By checking with the domain knowledge, we found that these genes are from one gene complex and are functional related. In another example, another group which contains gene PFY1 is dependent on the group with ARC19. The partial graph is shown in Figure 2. With the learned groups

and group Bayesian networks, a domain expert checked whether the learned results are consistent with domain knowledge. Many genes in the same groups are consistent with domain knowledge. New experiments are still on going for more genes.

## Application in a Changing Environment

In the proposed framework, a group Bayesian network is learned from data. It is the skeleton of the relationships among all the original variables in the domain and the structure among the group variables is more stable in different environment. Such a structure can be applied to some situations when the values of some original variables are missed. The values of the group variable can be inferred from the remaining original variables, without affecting the main structure of the group Bayesian network. In other cases, when the microarray data are from different experiments, the number of the original variables may not be the same. However, the functions in the cell and the groups of genes should be similar. Then the learned group Bayesian networks are comparable in this case.

## Discussion and Future Work

In this work, a new framework is proposed to learn Bayesian network from changing, multiple-source data with many variables. Group variables are introduced in this work to reduce the dimension of the data, speed up the learning process and generalize the Bayesian network for different experiment data. The proposed method can discover groups in variables and can identify the possible dependencies among groups. The observations in microarray data are used as a special background for this method.

There are some assumptions in the proposed framework. One assumption is that there are groups among the original variables – We assume that the connections among variables in a group are dense and the connections among groups should be sparse. And the other assumption is that group variables can represent the group of original variables reliably in different conditions and the number of members in each group can change according to the environment. By introducing group variables and restricting the links among variables in the different groups, our methods will lose some details in the sense of the direct interactions among genes. However, it can capture the main interactions among groups. This is a high-level abstraction of interactions among gene functions and it can speed up the learning process. This type of abstraction is common in biology. If it is possible to extract partial knowledge of the grouping and the group interaction from domain knowledge, the learning process will be speeded up further. For other domains, if the relationships among the variables are satisfied with the above assumptions, our framework is also applicable. For

example, the data from stock market is another application. The different sections in stock market can be treated as different groups in our framework and our framework may determine certain relationships among different sections.

Some other research efforts are related to the proposed method in microarray data analysis domain. The similarities and differences between these works and the proposed method are discussed below. The first related work is clustering of microarray data set – which is a commonly used method in microarray data analysis. Clustering methods can identify genes with certain levels of similarity. However, clustering methods cannot identify the dependency and possible causal relationships among genes. This is not enough for biological knowledge discovery. In our proposed framework, we can identify the groups of genes and dependency among genes simultaneously. This is a better way to model the gene relationships.

The second related work is hidden variables discovery in Bayesian network. The general method for hidden variable detection is to identify hidden variables by maximal cliques or semi-maximal cliques in Bayesian network. The disadvantage of the general method is that it is difficult to identify the meaning of hidden variables. In our proposed framework, the hidden variables are assumed to represent the activity level of the functions of the gene groups.

The third related work is module network (Segal et al., 2003). In this work, the authors considered as modules groups of genes whose expression activity levels are similar. The authors assumed that the variables in the same modules have the same parents and the same conditional probability distributions. This assumption is one type of parameter tying in Bayesian networks and can narrow the structure space and the number of parameters in Bayesian network learning process. But the authors in (Segal et al., 2003) did not consider group variables. The Bayesian network in their work was learned with the original variables. So the search space of Bayesian network structure is still very large.

One more related work is the first-order probabilistic model. Now the research in the first-order probabilistic model is to build a model from domain knowledge and do inference after instantiating the model for a specific case. In our work, the group variables can be treated as population variables in the first-order probabilistic model and each member in the groups can be treated as an individual of a population. Then our work provides a way to learn a first-order probabilistic model from data. And the relationships among group variables and original variables are well measured in our model and the inference can be done with probabilistic meaning other than instantiation in the first-order probabilistic model.

The on-going and future agenda are as follows. One is to do overlapped grouping. In the current algorithm, each gene is only assigned to one group. However, from biological knowledge, we know some genes can perform several functions and belong to different groups.

Overlapped grouping is a natural way to model this phenomenon.

Another important future work is to collaborate with domain experts. Biologists in a local institute are doing research on the function of genes in Actin cytoskeleton group of yeast cell. The proposed method provides them computational support for hypotheses of dependency among genes. A domain expert has checked the learned dependencies among groups of genes from our method. Many of the group assignments and the group dependencies are consistent with the domain knowledge. The expert also chose some dependencies among genes as reasonable hypotheses in his biological experiments. The real biological experiment is still ongoing and we will verify our method with the real biological result later.

## Acknowledgment

We would like to thank Mr. Qiu Wenjie for checking the learned Bayesian network with yeast genes in Actin cytoskeleton group. This research is supported by Research Grant No. BM/00/007 from the Biomedical Research Council (BMRC) of the Agency for Science, Technology, and Research (A\*Star) and the Ministry of Education in Singapore.

## References

- Cheeseman, P., & Stutz, J. (1996). Bayesian classification (autoclass): Theory and results. In U. M. Fayyad, G. Diatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining* (pp. 153-180): AAAI Press/The MIT Press.
- Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3, 507-554.
- Chickering, D. M., & Heckerman, D. (1997). Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Machine Learning*, 29, 181-212.
- Cooper, G. F., & Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309-347.
- Dash, D. H., & Druzdzel, M. J. (1999, Jul 30-31, 1999). A hybrid anytime algorithm for the construction of causal models from sparse data. Paper presented at the Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99), Stockholm, Sweden.
- Friedman, N. (1998). The bayesian structural em algorithm. Paper presented at the Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000) Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7 (3-4). 601-620.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., et al. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11(12), 4241-4257.

JOSHI, R. & LEONG, T.-Y. (2005) Adaptive Modeling Framework for Systems Biology. To appear in Proceedings of the 2005 AAAI Spring Symposium on Challenges to Decision Support in a Changing World. Stanford University, CA, USA.

Lu, T.-C., Druzdzel, M. J., & Leong, T.-Y. (2000). Causal mechanism-based model constructions. Paper presented at the Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence, Stanford University, Stanford, California, USA.

Madigan, D., & York, J. (1995). Bayesian graphical models for discrete data. *International statistical Review*, 63, 215-232.

Murphy, K. (2004). Software packages for graphical models / bayesian networks,  
<http://www.Ai.Mit.Edu/~murphyk/software/bnt/bnsoft.html>

Neapolitan, R. E. (2004). *Learning bayesian networks*: Prentice Hall.

Pearl, J., & Verma, T. (1991). A theory of inferred causation. Paper presented at the Principles of Knowledge Representation and Reasoning: Proceeding of the Second International Conference, San Mateo, CA.

Segal, E., Pe'er, D., Regev, A., Koller, D., & Friedman, N. (2003). Learning module networks. Paper presented at the Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI), Acapulco, Mexico.

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search*. Number 81 in lecture notes in statistics. New York: Springer Verlag.