Using a Dynamic Bayesian Network to Learn Genetic Interactions

Linus Göransson
Graduate School of Biomedical Research
Linköping University
lingo557@student.liu.se

Timo Koski
Department of Mathemathics
Linköping University
tikos@mai.liu.se

Abstract

In this paper, we evaluate a method for analyzing microarray data. The method is an attempt to learn regulatory interactions between genes from gene expression data. It is based on a Bayesian network, which is a mathematical tool for modeling conditional independences between stochastic variables. We review the dynamic nature of interacting genes, and explain how to model them using such a network. We also discuss an algorithm for learning the structure of interacting genes from gene expression data, using scaled mutual information for scoring. Finally, the method is applied to data from stress response experiments in Yeast. Although no biologically significant findings were made here, we believe the method has future potential.

1 Introduction

The gene sequences contained in our genome has all the information needed to make cells grow, differentiate, and respond to environmental changes. During this process, genes do not remain static. Instead, they constantly fluctuate, being turned on at one moment and turned off at the next. This dynamic process is regulated by a complex network of interacting genes, where individual genes affect, and in turn are affected by, the state of many other genes. The analysis of these fluctuations in expression levels was made much easier with the recent introduction of microarray technology. Using this method, expression levels of thousands of genes can be measured simultaneously, as they change over time and are affected by different stimuli. Thereby, it is possible to obtain a global view of the dynamic interaction between genes.

This type of experiments generates large amount of raw data, typically too large to be analyzed manually. Therefore, much effort is currently being put into designing automatic analytical methods for microarray data. One widely used method is the use of different clustering algorithms. These algorithms are capable of grouping genes with similar expression patterns together. The method is based on the assumption, that genes with similar expression patterns also have similar functions. From this assumption it is possible to assign unknown genes with a function by looking at the genes it is clustered with.

A more challenging problem than assigning functions to individual genes is to try and discover the network of interacting genes that generate the fluctuations in expression levels. Obtaining a map of this genetic network would give a whole new perspective on genetic interactions, leading to new insights in the regulation of genes. Recently, there has been a lot of interest in the concept of *reverse engineering* of genetic networks, i.e. inferring the underlying genetic network from analysis of the fluctuations in gene expression levels. In this research, the use of Bayesian networks has emerged as a very promising method.

A Bayesian network (BN) is a statistical tool, that for the last decade has become popular in the areas of machine learning and artificial intelligence (Cowell *et al*. 1999). Bayesian networks are well suited for inferring genetic interactions because of their ability to model *causal influence* (cause - effect) between variables (in this case, genes). BNs are also capable of handling the stochastic nature of gene expression data, such as noise, hidden variables, and missing data. Finally, BNs are capable of extracting knowledge from data sets with too few replicates for conventional statistics to be reliable (Spirtes *et al.* 2000). This lack of experimental replication is a common issue when analyzing gene expression data.

In this paper, we use a Bayesian network learning algorithm to analyze microarray data. The algorithm is first tested on simulated data to evaluate its performance. After testing, the algorithm is used to infer a genetic

network structure from experimental data. The reminder of this paper will be organized as follows: In section 2, the dynamics of gene expression is discussed. In section 3, a mathematical model that captures the dynamic properties of genetic networks is proposed. In section 4, the implementation of the model as a dynamic BN is described. In section 5, results from applying the dynamic BN on synthetic and experimental data are presented. We conclude with a discussion of the method used in this paper, possible extensions to it, and a summary of related work in section 6.

2 Dynamics of gene expression

There are several ways for a gene to affect the expression level of another gene. When mRNA has been transcribed from an active gene, it is transported to the ribosomes, where it is translated into a specific protein. Proteins are the main effector molecules in the cell, capable of a wide range of functions. It is on the protein level that most of the regulatory interactions occur. These interactions will in turn have effect on the expression levels of the underlying genes.

Activation of a gene will quickly result in increased levels of mRNA from that gene. Typically, mRNA levels will start rising within 10 min from stimulation (e.g. rapid heating) of the cell. This time is required for the stimulation to activate various transcription factors that in turn will bind to, and direct the transcription of the specific genes. When these genes has become active, they will affect the transcription of many other genes that in *their* turn will affect others in a time-dependent manner. In figure 1, this dynamic behavior is shown for some genes following hydrogen peroxide treatment in Yeast.

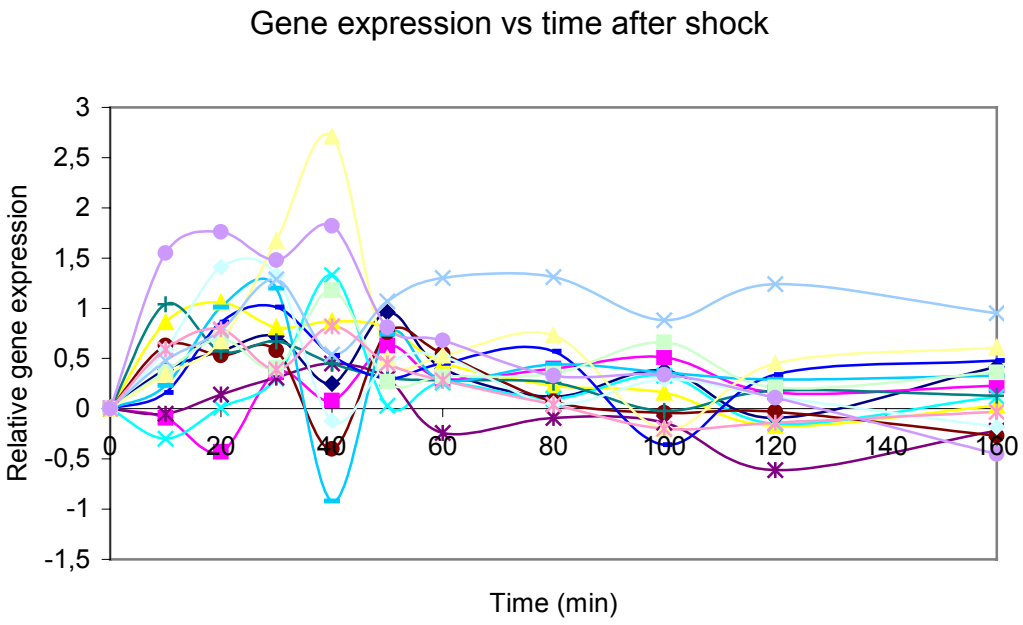

Figure 1. The dynamic behavior of gene expression for 15 genes, following hydrogen peroxide treatment in Yeast. The relative expression levels are set to zero at time zero. The measured values have been connected with fitted curves for clarity. Data are from (Gash *et al*. 2000).

The dynamic nature of the gene expression data suggests that expression levels at one time are functions of the expression levels at previous time-points. If this is true, it may be possible to use measured expression data to infer the underlying network of interactions between the genes.

3 Genetic network modeling

We will use a dynamic Bayesian network for representing and learning the structure of genetic networks from gene expression data. A BN consists of two parts. First, there is a directed acyclic graph (DAG), representing random variables as nodes, and relationships between variables as arcs between the nodes. Second, there is a set

of parameters for each of the nodes in the DAG that together specify a joint distribution over the random variables. The joint distribution is factorized along the graph using assumptions of conditional independence. A *dynamic* BN extends the static BN by modeling changes of stochastic variables over time. In this paper, it is presumed the reader is familiar with the theory behind Bayesian networks and graphical models. For those unfamiliar with the subject, (Heckerman 1996) gives a compact introduction, while (Jensen 1996) is a good introductory textbook.

Assuming that expression levels at one time are functions of the expression levels at previous time-points, we are able to model the dynamics of genetic networks. The gene expression data we are analyzing in this paper are organized in an N x T matrix form, with n genes, measured at T time-points. Each gene expression value is indexed by the gene number n and the time-point t, where t is an integer. We use $\mathbf{X}[t] = \{X_1[t],\ldots,X_N[t]\}$ to denote a vector of stochastic variables representing genes 1 to N, measured at time-point t. A complete set of expression data consists of T such vectors, $\mathbf{X}[1],\ldots,\mathbf{X}[T]$, together forming the N x T matrix. The vector $\mathbf{X}[t]$ can be seen as a time-slice from the complete data matrix $\mathbf{X}[1],\ldots,\mathbf{X}[T]$.

We mentioned above, that expression levels at one time are functions of the expression levels at previous time-points. If this is implemented in our model, a time-slice $\mathbf{X}[t+1]$ would be dependent of the values of all previous time-slices $\mathbf{X}[1],\ldots,\mathbf{X}[t]$. Although this is the biologically most realistic model, it will be computationally very costly. A simplified model, where expression levels at one time are functions only of the most recent time-point is therefore used. In this model, time slice $\mathbf{X}[t+1]$ is dependent only of time-slice $\mathbf{X}[t]$. This property is known as the *Causal Markov Assumption*: given the values of a variable's immediate causes, it is independent of its earlier causes. In figure 2, this model is represented as a dynamic BN.

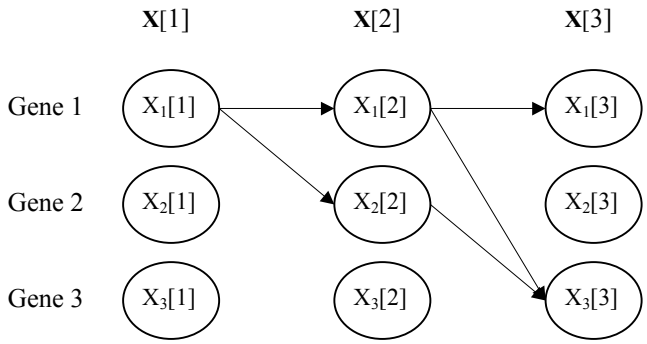

Figure 2. A gene network model, represented as a dynamic Bayesian network. In this model, expression levels for three genes are included; each measured at three different time-points. Circles are nodes, representing stochastic variables, and arcs between nodes represent conditional independences between variables. In this model, nodes in time slice X[t+1] are dependent only of nodes in time-slice X[t].

Assuming that the expression level of a gene is a function only of expression levels of other genes is appealing, because it makes it possible to build a relatively simple model of the interacting genes. Of course, this assumption is not true. As mentioned in the previous section, genes do not interact directly with each other. Instead, they do so by means of mediating proteins and other molecules. Therefore, the proposed model is a very simplified model of the genetic interactions. Mediating agents should be represented by the introduction of hidden variables (variables we cannot observe). Introducing hidden variables is quite possible in BNs, but not with the learning algorithms we are using in this paper. Therefore, the simpler model has been used hoping it will still give meaningful results.

4 Learning the network

Learning the structure of a dynamic BN means the following: Given a training set of data, find the network structure that best fits the data. The network we want to learn is the *transition network,* i.e. the network defining dependencies between the adjacent time-slices $\mathbf{X}[t]$ and $\mathbf{X}[t+1]$. The transition network for our proposed model is shown in figure 3. The training set of data consists of all adjacent time-slices $\mathbf{X}[t]$ and $\mathbf{X}[t+1]$ in our data set.

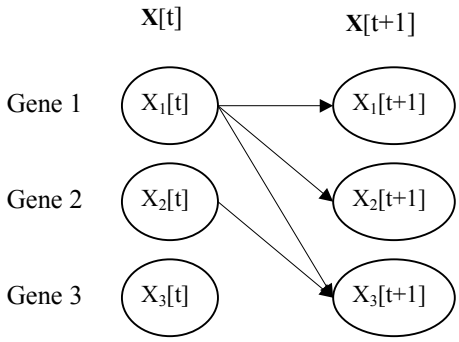

Figure 3. The transition network for the genetic network in figure 2. The transition network defines dependencies between the adjacent time-slices X[t] and X[t+1].

The algorithm we are using to infer network structure from training data is called REVEAL (Liang *et al.*1998). It is based on the information theoretic concept of *mutual information analysis* of the data. The algorithm learns the optimal set of parents for each node independently, by choosing the parent set that maximizes a scoring function. This scoring function is defined by I(X, Pa(X))/max{H(X), H(Pa(X))}, where I(X, Y) is the mutual information between X and Y, and H(X) is the entropy of X. For definitions, see (Murphy and Mian, 1999). For another similar application, see (Gyllenberg and Koski 2002).

For a large number of nodes N, it is impossible to calculate the score for all combination of parents, as there are Σ k=0 to N (N over k) = 2^N such combinations for each of the nodes. The solution to this problem is placing a limit on the maximum number of parent each node is allowed to have, defined as the max fan-in K. This limits the number of parent combinations to Σ k=0 to K (N over k), which is computable for a sufficiently low K.

5 Implementation

The model for analyzing gene expression data was implemented using the mathematical programming language MatLab. MatLab is a standard tool at universities and many industries, especially within the engineering community. The main strength of MatLab is the many excellent toolboxes for various areas such as statistics, optimization, etc, which are readily available. In fact, one such toolbox, the Bayes Net Toolbox (BNT) (Murphy 2001), was used for implementing the Bayesian network. BNT is an open-source MatLab package for directed graphical models. It supports many types of nodes (probability distributions), exact and approximate inference, parameter and structure learning, and static as well as dynamic models. BNT can be downloaded from http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html

In order to analyze expression data, it was first transformed from continuous to discrete form. The number of discrete steps to be used is arbitrary, but for computational reasons it should be kept as low as possible. Using three steps, one for unchanged, one for up-regulated, and one for down-regulated expression levels, is the natural choice.

The actual structure learning was performed by calling the BNT function learn_struct_dbn_reveal, which uses the REVEAL algorithm discussed in the previous section. The resulting inter-slice adjacency matrix, which defines the transition network, was then visualized using another BNT function, draw_layout. Upon testing the script, it was found that the number of nodes had to be limited to 15, as more nodes was too computationally demanding.

6 Applying the model on data

6.1 Synthetic data

In order to evaluate the learning algorithm, it was tested using artificial data. The data was generated from an idealized genetic network, containing 5 interconnected nodes and 10 additional, independent nodes. The structure of this synthetic network is shown in figure 4.

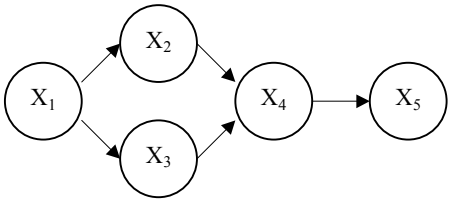

Figure 4. An idealized genetic network, containing 5 interconnected nodes and 10 additional, independent nodes (not shown). This structure was used for generating artificial genetic data.

The nodes were given random conditional probability distributions over the three states corresponding to unchanged, down- and up-regulated, respectively. Training data were then generated by sampling instances from this artificial genetic network. Once a large set of training data had been created the REVEAL algorithm was used to find the structure of the correct network. Finally, the number of incorrect edges in the learned network was calculated for different sizes of training data sets. The resulting graph is shown in figure 5.

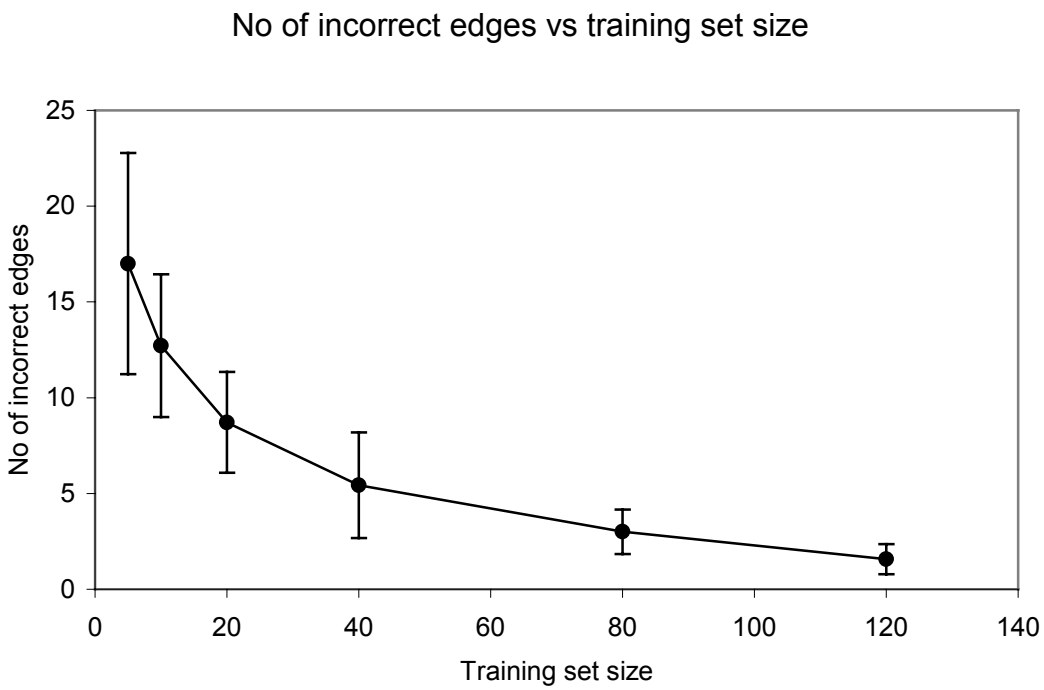

Figure 5. The number of incorrect edges in the learned network, calculated for different sizes of training data sets. The bars represent the standard deviation of the number of incorrect edges.

It seems as if the number of incorrect edges approaches zero asymptotically, as the size of the training data increases. For a data training data size of 160 time-slices, there are on average 1.5 incorrect edges. Thus, it seems to be possible to obtain a good approximation of the network structure with a realistic amount of training data.

6.2 Biological data

For this study, we used publicly available data from the Yeast stress experiments of (Gash *et al.* 2000). In these experiments, the expression patterns in the Yeast *S. cervisiae* was explored during the response to various environmental transitions. DNA microarrays were used to measure changes in transcript levels as cells responded to temperature shocks, hydrogen peroxide, the drug menadione, the sulfhudryl-oxidizing agent

diamide, the disulfide-reducing agent dithiothretiol, hyper- and hypo-osmotic shock, amino-acid starvation, nitrogen source depletion, and progression into stationary phase. A large set of genes, consisting of approximately 900 genes, showed similar drastic responses to the treatments.

There were a number of reasons for choosing this data set. First of all, it was a comparatively large data set, consisting of 76 measurements. Second, the measurements were from 11 different treatments, that all induce a similar set of genes, although in a slightly different manner. This is ideal, if the goal is to infer the underlying genetic network structure from the data. Finally, all the measurements were from time-series, which is required for using the REVEAL algorithm. Stress responses are usually fast events, and most of the measurements are therefore made with short intervals between them. This ensures that no interactions are lost because they occur between two measurements.

From this data set, a subset consisting of 15 genes was chosen as training data. The subsets of genes were chosen on the basis that they are all involved in intracellular signaling. This was an arbitrary choice, based on the idea that genes involved in intracellular signaling may be involved in many interesting interactions. The genes are listed in table 1. In the transformation to discrete values, the threshold value for up- and down-regulated genes were set to >1 and <-1, respectively.

Gene	Description
PKA3	CAMP-DEPENDENT PROTEIN KINASE SUBUNIT
PDE1	PHOSPHODIESTERASE
IKS1	PROTEIN KINASE
YAK1	SER/THR PROTEIN
TOR1	PHOSPHATIDYLINOSITOL 3-KINASE
MTL1	PUTATIVE KINASE THAT ACTS WITH MID2 CELL WALL SENSOR
PIG2	PUTATIVE GLC7 REGULATORY SUBUNIT
SIP2	COMPONENT OF SNF4 COMPLEX
RIM11	PROTEIN KINASE
PTK2	PUTATIVE PROTEIN KINASE
KNS1	PUTATIVE SER/THR PROTEIN KINASE
KIN82	PUTATIVE SER/THR PROTEIN KINASE
YMR251W	SIMILAR TO SER/THR PROTEIN KINASE
YIL113W	SIMILAR TO PROTEIN PHOSPHATASE
RRD2	SIMILAR TO HUMAN PROTEIN PHOSPHATASE ACTIVATOR PROTEIN

Table 1. The training data set consisting of 15 genes, chosen on the basis that they are all involved in intracellular signaling

The resulting graph, after applying the REVEAL algorithm on the data, is shown in figure 6. The algorithm found seven different interactions between the genes. The three genes that appeared to have modulating functions were PKA3, YAK1, and KNS1. The suggested interactions were examined manually by searching the PUBMED article database for co-occurrences, but no support for them was found in literature. Many of the interacting genes have in common, that they are involved in the cell cycle regulation. It should be kept in mind, however, that the genes were selected in the first place for being affected in shock-response, and for being involved in intracellular signaling. These selection criterions will both favor cell cycle regulating genes.

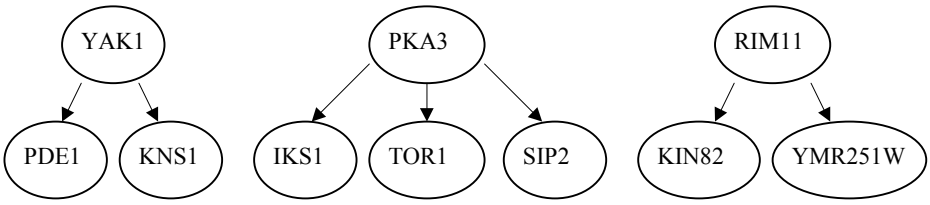

Figure 6. The resulting genetic network, after applying the REVEAL algorithm on a biological data set consisting of 15 genes. The three genes that appear to have modulating functions are PKA3, YAK1, and RIM11.

7 Discussion

In this paper, we have tried to infer the structure of a genetic network from analysis of gene expression data. We have modeled the interacting genes as a dynamic Bayesian network, and used the REVEAL algorithm for learning the structure of these. We then evaluated the method on both synthetic and biological data.

The results from the artificial data set confirmed that the method is capable of learning the correct structure of a genetic network, given enough training data. It should be noted, however, that computer-generated data from idealized genetic networks are not equal to real-life, biological data. Any estimation of the required size of training data based upon artificial data should therefore be treated with care. Upon analyzing biological data, the algorithm was able to learn a network for the gene interactions. Unfortunately, no support for these proposed interactions was found in literature.

The results indicate that the method may work in theory, but that there are a number of issues that have to be solved before it can be used in practice. First, the current algorithm was too computationally heavy. Currently, it can analyze up to 15 genes simultaneously. This is obviously not enough – in the data used in this paper, about 900 genes were affected by the various treatments. The solution to this problem is finding a more efficient search algorithm. A promising candidate is the *sparse candidate* algorithm (Friedman *et al.* 2000). The main idea behind this algorithm is to identify a relatively small number of candidate parents for each gene based on simple local statistics. The search for the optimal set of parents can then be performed inside this smaller search space using the standard mutual information approach. This algorithm has been successfully used on a data set consisting of 800 genes. The model is based on a static BN rather than a dynamic BN, although there are plans on extending it to a dynamic model.

Other problem concerns the experimental design: How many measurements are needed for structure learning to be reliable? How many different treatments should be included? What is the optimal interval between measurements? There are currently no published analytical results on this subject. There is also the issue of interventional data, where the experimenter manipulates individual variables and studies the affect of this (Murphy 2001). In a genetic network, this can be achieved by creating knockout / overexpression mutants. Using interventional data can be a very powerful tool for learning genetic networks, but it requires careful planning of the experimental design to be informative.

Alhough we were unable to extract meaningful biological information with our current methods, we feel there is great potential in applying Bayesian networks on gene expression data. Hopefully, the advances made in the area together with more powerful computers will soon result in the discovery of new genetic regulatory networks.

8 Acknowledgements

This work was partly funded by the Foundation for Strategic Research, and was carried out as a project within the Graduate School of Biomedical Sciences, University of Linköping.

9 Bibliography

Cowell, R.G., David, A.P., Lauritzen, S.L., Spiegelhalter D.J. 1999. Probabilistic Networks and Expert Systems. Springer Verlag, Statistics in Engineering and Informations Scinence, New York.

Friedman, N., Murphy, K., Russell, S. Learning the Structure of Dynamic Probabilistic Networks. 1998. Proc UAI, 139-147.

Friedman, N., Linial, N., Nachman, I., Pe'er, D. 2000. Using Bayesian networks to analyze expression data. J Comp Biol 7, 601-620.

Gasch, A.p., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., Brown, P.O. Genomic expression program in the response of yeast cells to environmental changes. Mol. Bio. Cell, 11, 4241-4257.

Gyllenberg, M., Koski, T. 2002. Tree Augmented Classification of Binary Data Minimizing Stochastic Complexity. Technical Report, Department of Mathematics, University of Linköping.

Heckerman, D. 1996. A Tutorial on Learning with Bayesian Networks. Microsoft Research Technical Report 95-06.

Jensen, F.V., 1996. An introduction to Bayesian networks. UCL Press, London.

Liang, S., Fuhrman, S., Somogyi, R. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. Pacific Symposium on Biocomputing. 3, 18-29.

Murphy, K. 2001. Active Learning of Causal Bayes Net Structure. Technical report, Computer Science Division, University of California, Berkeley, CA.

Murphy, K. 2001. The Bayes Net Toolbox for MatLab. Technical report, Computer Science Division, University of California, Berkeley, CA.

Murphy, K., Mian, S. 1999. Modelling Gene Expression Data using Bayesian Networks. Technical report, Computer Science Division, University of California, Berkeley, CA.

Spirtes, P., Clark Glymour, G., Scheines, R., Kauffman, S., Aimale, V., Wimberly, F. 2000. Constructing Bayesian Network Models of Gene Expression Networks from Microarray Data. Proc. of the Atlantic Symposium on Computational Biology.