Using Bayesian Networks to Analyze Expression Data

Nir Friedman
Institute of Computer Science
Hebrew University
Jerusalem, 91904, ISRAEL
nir@cs.huji.ac.il

Michal Linial
Institute of Life Sciences
Hebrew University
Jerusalem, 91904, ISRAEL
michall@leonardo.ls.huji.ac.il

Iftach Nachman
Center for Neural Computations
Hebrew University
Jerusalem, 91904, ISRAEL
iftach@cs.huji.ac.il

Dana Pe'er
Institute of Computer Science
Hebrew University
Jerusalem, 91904, ISRAEL
danab@cs.huji.ac.il

Abstract

DNA hybridization arrays simultaneously measure the expression level for thousands of genes. These measurements provide a "snapshot" of the cell's transcriptions. A major challenge in computational biology is to uncover, from such measurements, gene/protein interactions and key biological features of the cellular system.

In this paper, we propose a new framework for discovering interactions between genes based on multiple expression measurements. This framework builds on the use of *Bayesian networks* for representing statistical dependencies. A Bayesian network is a graphical model of joint multivariate probability distributions that captures properties of conditional independence between variables. Such models are attractive for their ability to describe complex stochastic processes, and for providing clear methodologies for learning from (noisy) observations.

We start by showing how Bayesian networks can describe interactions between genes. We then present an efficient algorithm capable of learning such networks and a statistical method to assess our confidence in their features. Finally, we apply this method to the *S. cerevisae* cell-cycle measurements of Spellman et al. (1998) to uncover biological features.

1 Introduction

A central goal of molecular biology is to understand the regulation of protein synthesis and its reactions to external and internal signals. All the cells in an organism carry the same genomic data, but their protein makeup can be drastically different both temporally and spatially. Protein synthesis is regulated by many mechanisms at its different levels. These include mechanisms for controlling transcription initiation, RNA splicing, mRNA transport, translation initiation, post-translational modifications, and degradation of mRNA/protein. One of the main junctions at which regulation occurs is mRNA transcription. A major role in this machinery is played by proteins themselves, that bind to regulatory regions along the DNA, greatly affecting the transcription of the genes they regulate.

In recent years, technical breakthroughs in spotting hybridization probes and advances in genome sequencing efforts lead to development of *DNA microarrays*, which consist of many species of probes, either oligonucleotides or cDNA, that are immobilized in a predefined organization to a solid surface. By using DNA microarrays researchers are now able to measure the abundance of thousands of mRNA targets simultaneously (DeRisi. et al. 1997, Lockhart et al. 1996, Wen et al. 1998). Unlike classical experiments, where the expression levels of only a few genes were reported, DNA microarray experiments can measure *all* the genes of an organism, providing a “genomic” viewpoint on gene expression. As a consequence, this technology facilitates new experimental approaches for understanding gene expression and regulation (Iyer et al. 1999, Spellman et al. 1998).

Early microarray experiments examined few samples, and mainly focused on differential display across tissues or conditions of interest. The design of recent experiments focuses on performing a larger number of microarray experiments ranging in size from a dozen to a few hundreds of samples. In the near future, data sets containing thousands of samples will become available. Such experiments collect enormous amounts of data, which clearly reflect many aspects of the underlying biological processes. An important challenge is to develop methodologies that are both statistically sound and computationally tractable for analyzing such data sets and inferring biological interactions from.

Most of the analysis tools currently in use are based on *clustering* algorithms. These algorithms attempt to locate groups of genes that have similar expression patterns over a set of experiments (Alon et al. 1999, Ben-Dor & Yakhini 1999, Eisen et al. 1998, Michaels et al. 1998, Spellman et al. 1998). Such analysis is useful in discovering genes that are co-regulated. A more ambitious goal for analysis is revealing the structure of the transcriptional regulation system (Akutsu et al. 1998, Chen et al. 1999, Somogyi et al. 1996, Weaver et al. 1999). This is clearly a hard problem: Mainly since mRNA expression data alone gives only a partial picture that does not reflect key events, such as translation and protein (in)activation, which play a major role in regulation of mRNA transcription. In addition, the amount of samples, even in the largest experiments in the foreseeable future, does not provide enough information to construct a full detailed model with high statistical significance. Finally, using current technology, even these few samples have a high noise to signal ratio, at times the noise being much stronger than the signal.

In this paper, we introduce a new approach for analyzing gene expression patterns, that uncovers properties of the transcriptional program by examining statistical properties of *dependence* and *conditional independence* in the data. We base our approach on the well-studied statistical tool of *Bayesian networks* (Pearl 1988). These networks represent the dependence structure between multiple interacting quantities (e.g., expression levels of different genes). Our approach, probabilistic in nature, is capable of handling noise and estimating the confidence in the different features of the network. We are therefore able to focus on interactions whose signal in the data is strong.

Bayesian networks are a promising tool for analyzing gene expression patterns. First, they are particularly useful for describing processes composed of *locally* interacting components; that is, the value of each component *directly* depends on the values of a relatively small number of components. Second, statistical foundations for learning Bayesian networks from observations, and computational algorithms to do so

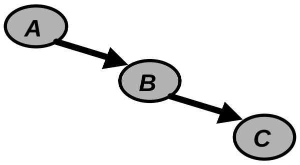

Figure 1: An example of a simple network structure.

are well understood and have been used successfully in many applications (Friedman et al. 1997, Thiesson et al. 1998). Finally, Bayesian networks provide models of causal influence: Although Bayesian networks are mathematically defined strictly in terms of probabilities and conditional independence statements, a connection can be made between this characterization and the notion of *direct causal influence*. (Heckerman et al. 1997, Pearl & Verma 1991, Spirtes et al. 1993).

The remainder of this paper is organized as follows. In Section 2, we review key concepts of Bayesian networks, learning them from observations, and using them to infer causality. In Section 3, we describe how Bayesian networks can be applied to model interactions among genes and discuss the technical issues that are posed by this type of data. In Section 4, we apply our approach to gene-expression data of Spellman et al. (1998), analyzing the statistical significance of the results and their biological plausibility. Finally, in Section 5, we conclude with a discussion of related approaches and future work.

2 Bayesian Networks

2.1 Informal Introduction

Before giving a formal definition of Bayesian Networks, we will first try to demonstrate the basic concept through several examples.

Let $P(X,Y)$ be a joint distribution over two variables X and Y. We say that variables X and Y are *independent* if $P(X,Y) = P(X)P(Y)$ for all values of X and Y. (Equivalently, $P(X|Y) = P(X)$.) Otherwise, the variables are *dependent*. When X and Y are dependent, learning the value Y gives us information about X. Note that correlation between variables implies dependence. However, dependent variables might be uncorrelated. (Formally, correlation is a sufficient but not a necessary condition for dependence.)

For concreteness, we now consider, a somewhat simplistic, biological example. Assume gene A is a transcription factor of gene B. Therefore, we expect their level of their expression to be dependent. For example, measuring high expression levels of gene A, we expect to find gene B over-expressed as well. Alternatively, gene A might be inhibiting the transcription of gene B, in which case over-expression of A implies under-expression of B.

We can represent such dependencies using a graph, in which each variable is denoted by a node. When two variables are dependent we draw an edge between them; see Figure 1. If the arrow points from A to B, we call A the *parent* of B.

We now consider slightly more complex situation that involves three genes A, B, and C. Suppose that gene B is a transcription factor of gene C. The expression levels of each pair of genes (i.e. A and B, A and C, and B and C) are dependent. If A does not directly affect C, then we should expect that once we fix the expression level of B (e.g., by knocking out B) we will observe that A and C are independent. In other words, the effect of gene A on gene C is *mediated* through gene B. Once we know the expression level of gene B, the expression of gene A does not give new information about the expression of gene C. In this case, we have

$$P(A \mid B, C) = P(A \mid B)$$

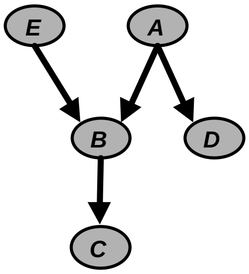

Figure 2: An example of a simple Bayesian network structure.
This network structure implies several conditional independence statements: $I(A;E)$, $I(B;D \mid A,E)$, $I(C;A,D,E \mid B)$, $I(D;B,C,E \mid A)$, and $I(E;A,D)$.
The network structure also implies that the joint distribution can be specified in the product form

$$P(A,B,C,D,E) = P(A)P(B|A,E)P(C|B)P(D|A)P(E)$$

and we say that A and C are *conditionally independent*, given B. We denote such a conditional independence as $I(A;C \mid B)$.

We want to represent such conditional independencies in our description of the interactions between the variables. In the graph representation, this is achieved by not having an edge between A and C, thus the dependency between them is represented as a directed path through B; see Figure 1.

Clearly, linear sequences of dependence are not the only type of dependencies. To see this, suppose our ongoing example involves another gene, D, that is also regulated by A. As before, all three pairs of genes are correlated. But, genes B and D are independent once we know the expression level of A. Using our notation: $I(B;D \mid A)$. Thus, gene A explains the dependence between B and D. In such a situation, we say that gene A is a *common cause* of genes B and D. We model this relation as shown in Figure 2. At this point it is interesting to note that if the expression of gene A is not measured, then B and D would appear dependent in data and we would have drawn an edge between them. In such a case we call A a *hidden common cause*.

Now suppose that gene E inhibits the transcription of gene B. We model this, by placing an arc from E to B; See Figure 2. In this case, the expression of B is regulated by two genes (A and E). These are B's *parents*, denoted as **Pa**(B) If the expression level of A is high, we expect B to be expressed as well, unless the expression of E is also high. In that case, we expect the expression of B to be low even though A is high. This leads us to the second component of a Bayesian Network. In addition to a graph that describes (in)dependencies between variables, each variable is described as a stochastic function of its parents. Specifically, we associate with each variable X a conditional probability model that specifies the probability of X given its parents. We denote the probability of a variable (gene) X to have the value (expression level) x given the values of its parents **pa**(X) as $P(x|\mathbf{pa}(X))$.

Using stochastic models is natural in the gene expression domain for several reason: First, the biological processes we want to model are stochastic. (Regardless of whether this is inherent stochasticity, or a function of our inability to measure some of the quantities that determine the exact expression levels.) Second, the measurements of the underlying biological system are noisy.

2.2 Representing Distributions with Bayesian Networks

We now review the formal definition of Bayesian networks. Consider a finite set $\mathcal{X} = \{X_1, \ldots, X_n\}$ of random variables where each variable X_i may take on a value x_i from the domain Val(X_i). In this paper, we focus on finite domains, though much of the following holds for infinite domains, such as continuous valued random variables. We use capital letters, such as X, Y, Z, for variable names and lowercase letters x, y, z to denote specific values taken by those variables. Sets of variables are denoted by boldface capital letters $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$, and assignments of values to the variables in these sets are denoted by boldface lowercase letters **x**, **y**, **z**. We denote $I(\boldsymbol{X};\boldsymbol{Y} \mid \boldsymbol{Z})$ to mean $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$ conditioned on $\boldsymbol{Z}$.

A *Bayesian network* is a representation of a joint probability distribution. This representation consists of two components. The first component, G, is a directed acyclic graph whose vertices correspond to the random variables $X_1, \ldots, X_n$. The second component describes a conditional distribution for each variable, given its parents in G. Together, these two components specify a unique distribution on $X_1, \ldots, X_n$.

The graph G represents conditional independence assumptions that allow the joint distribution to be decomposed, economizing on the number of parameters. The graph G encodes the *Markov Assumption*: Each variable X_i is independent of its non-descendants given its parents in G. Formally, we denote this as:

$$\forall i \ , \ I(X_i; \mathit{NonDescendnets}(X_i) \mid \mathbf{Pa}(X_i)) \tag{1}$$

where $\mathbf{Pa}(X_i)$ is the set of parents of X_i in G, and $\mathit{NonDescendnets}(X_i)$ are the non-descendents of X_i in G. By applying the chain rule of probabilities and properties of conditional independencies, any joint distribution that satisfies (1) can be decomposed in the *product form*

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \mathbf{Pa}(X_i)).$$

Figure 2 shows an example of a graph G, lists the Markov independencies it encodes, and the product form they imply.

To specify a joint distribution, we also need to specify the conditional probabilities that appear in the product form. This is the second component of the network representation. This component describes distributions $P(x_i | \mathbf{pa}(X_i))$ for each possible value x_i of X_i, and $\mathbf{pa}(X_i)$ of $\mathbf{Pa}(X_i)$. In the case of finite valued variables, we can represent these conditional distributions as tables. Generally, Bayesian networks are flexible and can accommodate many forms of conditional distribution, including various continuous models.

Given a Bayesian network, we might want to answer many types of questions that involve the joint probability (e.g., what is the probability of $X = x$ given observation of some of the other variables?) or independencies in the domain (e.g., are X and Y independent once we observe Z?). The literature contains a suite of algorithms that can answer such queries (see e.g. (Jensen 1996, Pearl 1988)), exploiting the explicit representation of structure in order to answer queries efficiently.

2.3 Equivalence Classes of Bayesian Networks

A Bayesian network structure G implies a set of independence assumptions in addition to the independence statements of (1). Let $\mathrm{Ind}(G)$ be the set of independence statements (of the form X is independent of Y given $\boldsymbol{Z}$) that hold in all distributions satisfying these markov assumptions These can be derived as consequences of (1).

More than one graph can imply exactly the same set of independencies. For example, consider graphs over two variables X and Y. The graphs $X \rightarrow Y$ and $X \leftarrow Y$ both imply the same set of independencies (i.e., $\mathrm{Ind}(G) = \emptyset$). We say that two graphs G and G' are *equivalent* if $\mathrm{Ind}(G) = \mathrm{Ind}(G')$.

This notion of equivalence is crucial, since when we examine observations from a distribution, we often cannot distinguish between equivalent graphs. Results of (Chickering 1995, Pearl & Verma 1991) show that we can characterize *equivalence classes* of graphs using a simple representation. In particular, these results establish that equivalent graphs have the same underlying undirected graph but might disagree on the direction of some of the edges. Moreover, an equivalence class of network structures can be uniquely represented by a *partially directed graph* (PDAG), where a directed edge $X \rightarrow Y$ denotes that all members of the equivalence class contain the edge $X \rightarrow Y$; an undirected edge X—Y denotes that some members of the class contain the edge $X \rightarrow Y$, while others contain the edge $Y \rightarrow X$. Given a directed graph G the PDAG representation of its equivalence class can be constructed efficiently (Chickering 1995).

2.4 Learning Bayesian Networks

The problem of learning a Bayesian network can be stated as follows. Given a *training set* $D = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ of independent instances of $\mathcal{X}$, find a network $B = \langle G, \Theta \rangle$ that *best matches* D. The common approach to this problem is to introduce a statistically motivated scoring function that evaluates each network with respect to the training data, and to search for the optimal network according to this score.

A commonly used scoring function is the *Bayesian scoring* metric (see (Cooper & Herskovits 1992, Heckerman et al. 1995) for complete description): $\text{Score}(G : D) = \log P(G \mid D) = \log P(D \mid G) + \log P(G) + C$ where C is a constant independent of G and $P(D \mid G) = \int P(D \mid G, \Theta) P(\Theta \mid G) d\Theta$ is the *marginal likelihood* which averages the probability of the data over all possible parameter assignments to G. The particular choice of priors $P(G)$ and $P(\Theta \mid G)$ for each G determines the exact Bayesian score. Under mild assumptions on the prior probabilities, this scoring metric is asymptotically consistent: Given a sufficiently large number of samples, graph structures that exactly capture all dependencies in the distribution, will receive, with high probability, a higher score than all other graphs (Barron & Cover 1991, Friedman & Yakhini 1996, Höffgen 1993). This means, that given a sufficiently large number of instances in large data sets, learning procedures can pinpoint the exact network structure up to the correct equivalence class.

Heckerman et al. (1995) present a family of priors, called *BDe priors*, that satisfy two important requirements: First, these priors are *structure equivalent*, if G and G' are equivalent structures they are guaranteed to have the same score. Second, the priors are *decomposable*. That is, the score can be rewritten as the sum $\text{Score}_{\text{BDe}}(G : D) = \sum_i \text{ScoreContribution}_{\text{BDe}}(X_i, \mathbf{Pa}(X_i) : D)$, where the contribution of every variable X_i to the total network score depends only on its own value and the values of its parents in G. These two properties are satisfied for BDe priors when all instances $\mathbf{x}^\ell$ in D are *complete*—that is, they assign values to all the variables in $\mathcal{X}$.

Once the prior is specified and the data is given, learning amounts to finding the structure G that maximizes the score. This problem is known to be NP-hard (Chickering 1996), thus we resort to heuristic search. The decomposition of the score is crucial for this optimization problem. A *local* search procedure that changes one edge at each move can efficiently evaluate the gains made by adding, removing or reversing a single edge. An example of such a procedure is a greedy hill-climbing algorithm that at each step performs the local change that results in the maximal gain, until it reaches a local maximum. Although this procedure does not necessarily find a global maximum, it does perform well in practice, when combined with multiple random restarts. Examples of other search methods that advance using one-edge changes include beam-search, stochastic hill-climbing, and simulated annealing.

2.5 Learning Causal Patterns

A Bayesian network is a model of dependencies between multiple measurements. We are also interested in modeling the process that generated these dependencies. Thus, we need to model the flow of causality in the system of interest (e.g., gene transcription). A *causal network* is a model of such causal processes. It's representation is similar to a Bayesian network (i.e. a DAG where each node represents a random variable along with a local probability model for each node), the difference being it interprets the parents of a variable as its *immediate causes*.

We can relate causal networks and Bayesian networks, by assuming the *Causal Markov Assumption*: given the values of a variable's immediate causes, it is independent of its earlier causes. When the causal Markov assumption holds, the causal network satisfies the Markov independencies of the corresponding Bayesian network, thus allowing us to treat causal networks as Bayesian networks. This assumption is a natural one in models of genetic pedigrees: once we know the genetic makeup of the individual's parents the genetic makeup of her earlier ancestors are not informative about her own genetic makeup.

The main difference between causal and Bayesian networks, is that a causal network models not only the distribution of the observations, but also the effects of *interventions*. If X causes Y, then manipulating the value of X (i.e., setting it to another value in such a way that the manipulation itself does not affect the other variables), affects the value of Y. On the other hand, if Y is a cause of X, then manipulating X will not affect Y. Thus, although the Bayesian networks $X \rightarrow Y$ and $X \leftarrow Y$ are equivalent, as causal networks they are not.

When can we learn a causal network from observations? This issue received a thorough treatment in the literature (Heckerman et al. 1997, Pearl & Verma 1991, Spirtes et al. 1993). We only sketch the main idea here. From observations alone, we cannot distinguish between causal networks that specify the same independence assumptions, i.e., up to an equivalence class. Thus, as in Bayesian networks, we can only narrow down our model to an equivalence class. When learning an equivalence class (PDAG) from the data, we can conclude that the true causal network is possibly any one of the networks in this class. If a directed edge $X \rightarrow Y$ is in the PDAG, then all the networks in the equivalence class agree that X is an immediate cause of Y. Thus, we infer the causal direction of the interaction between X and Y.

3 Applying Bayesian Networks to Expression Data

In this section we describe our approach to analyzing gene expression data using Bayesian network learning techniques. We model the expression level of each gene as a random variable. In addition, other attributes that affect the system can be modeled as random variables. These can include a variety of attributes of the sample, such as experimental conditions, temporal indicators (i.e., the time/stage that the sample was taken from), background variables (e.g., which clinical procedure was used to get a biopsy sample), and exogenous cellular conditions.

By learning a Bayesian network based on the statistical dependencies between these variables, we can answer a wide range of queries about the system. For example, does the expression level of a particular gene depends on the experimental condition? Is this dependence direct, or indirect? If it is indirect, which genes mediate the dependency? We now describe how one can learn such a model from the gene expression data. Many important issues arise when learning in this domain. These involve statistical aspects of interpreting the results, algorithmic complexity issues in learning from the data, and preprocessing of the data.

Most of the difficulties in learning from expression data revolve around the following central point: Contrary to most previous applications of learning Bayesian networks, expression data involves transcript levels of thousands of genes while current data sets contain at most a few dozen samples. This raises problems in computational complexity and the statistical significance of the resulting networks. On the positive side, genetic regulation networks are sparse, i.e., given a gene, it is assumed that no more than a few dozen genes directly affect its transcription. Bayesian networks are especially suited for learning in such sparse domains.

3.1 Representing Partial Models

When learning models with so many variables, such small data sets are not sufficiently informative to significantly determine that a single model is the "right" one. Instead, many different networks should be considered reasonable given the data.[1] Our approach is to analyze this set of plausible networks. Although this set can be very large, we might attempt to characterize *features* that are common to most of these networks, and focus on learning them. Before we examine the issue of inferring such features, we briefly describe two classes of features involving pairs of variables. While at this point we focus only on pairwise features, it is clear that this analysis is not restricted to them.

The first type of features is *Markov relations*: Is Y in the *Markov blanket* of X? The Markov blanket of X is the minimal set of variables that *shield* X from the rest of the variables in the model. More precisely, X

[1] This observation is not unique to Bayesian network models. It equally well applies to other models that are learned from this data, such as clustering models.

given its Markov blanket is independent from the remaining variables in the network. It is easy to check that this relation is symmetric: Y is in X's Markov blanket if and only if there is either an edge between them, or both are parents of another variable (Pearl 1988). In the context of gene expression analysis, a Markov relation indicates that the two genes are related in some joint biological interaction or process. Note, two variables in a Markov relation are directly linked in the sense that no variable **in the model** mediates the dependence between them. It remains possible that an unobserved variable (e.g., protein activation) is an intermediate factor in their interaction.

The second type of features is *order relations*: Is X an ancestor of Y in all the networks of a given equivalence class? That is, does the given PDAG contain a path from X to Y in which all the edges are directed? This type of feature does not involve only a close neighborhood, but rather captures a long range property. Under the proper assumptions (see Section 2.5), learning that X is an ancestor of Y in the PDAG would imply that X is a cause of Y. However, these assumptions do not necessarily hold in the context of expression data. Thus, we view such a relation as an indication that X might be a causal ancestor of Y.

3.2 Estimating Statistical Confidence in Features

We now face the following problem: To what extent does the data support a given feature? More precisely, we want to estimate a measure of confidence in the features of the learned networks, where “confidence” approximates the likelihood that a given feature is actually true (i.e. is based on a genuine correlation and causation). Ideally, we would want to compute the posterior $P(G \mid D)$ over network structure. This would allow us to compute the posterior belief in each feature, by summing the posterior probability of all networks that have the feature. Unfortunately, we cannot compute the posterior explicitly since the number of possible networks is huge. Instead, we resort to an approximate method in the general spirit of the ideal solution.

An effective and relatively simple approach for estimating confidence is the *bootstrap* method (Efron & Tibshirani 1993). The main idea behind the bootstrap is simple. We generate “perturbed” versions of our original data set, and learn from them. In this way we collect many networks, all of which are fairly reasonable models of the data. These networks show how small perturbations to the data can affect many of the features.

In our context, we use the bootstrap as follows:

- For $i = 1 \ldots m$ (in our experiments, we set $m = 200$).
  - Re-sample with replacement, N instances from D. Denote by D_i the resulting data set.
  - Apply the learning procedure on D_i to induce a network structure $\hat{G}_i$.
- For each feature f of interest calculate confidence$(f) = \frac{1}{m}\sum_{i=1}^{m} f(\hat{G}_i)$, where $f(G)$ is 1 if f is a feature in G, and 0 otherwise.

We refer the reader to (Friedman, Goldszmidt & Wyner 1999) for more details, as well as large-scale simulation experiments with this method. These simulation experiments show that features induced with high confidence are rarely false positives, even in cases where the data sets are small compared to the system being learned. This bootstrap procedure appears especially robust for the Markov and order features described in Section 3.1.

3.3 Efficient Learning Algorithms

In Section 2.4, we formulated learning Bayesian network structure as an optimization problem in the space of directed acyclic graphs. The number of such graphs is super-exponential in the number of variables. As we consider hundreds and thousands of variables, we must deal with an extremely large search space. Therefore, we need to use (and develop) efficient search algorithms.

To facilitate efficient learning, we need to be able to focus the attention of the search procedure on relevant regions of the search space, giving rise to the *Sparse Candidate* algorithm (Friedman, Nachman & Pe'er 1999). The main idea of this technique is that we can identify a relatively small number of *candidate* parents for each gene based on simple local statistics (such as correlation). We then restrict our search to networks in which only the candidate parents of a variable can be its parents, resulting in a much smaller search space in which we can hope to find a good structure quickly.

A possible pitfall of this approach is that early choices can result in an overly restricted search space. To avoid this problem, we devised an iterative algorithm that adapts the candidate sets during search. At each iteration n, for each variable X_i, the algorithm chooses the set $C_i^n = \{Y_1, \ldots, Y_k\}$ of variables which are the most promising *candidate parents* for X_i. We then search for B_n, an optimal network in which $\mathbf{Pa}^{G_n}(X_i) \subseteq C_i^n$. The network found is then used to guide the selection of better candidate sets for the next iteration. We ensure that B_n monotonically improves in each iteration by requiring $\mathbf{Pa}^{G_{n-1}}(X_i) \subseteq C_i^n$. The algorithm continues until there is no change in the candidate sets.

We briefly outline our method for choosing C_i^n. In the initial phase of the algorithm, we use the score $\text{ScoreContribution}_{\text{BDe}}(X_i, X_j : D)$ to measure the quality of having X_j as a parent of X_i. We then set C_i^0 to be the k variables with the highest such scores. Since the scores of families are not additive, this choice of candidates can be sub-optimal. In later iterations, we take into account the network found in the previous iteration, and measure the quality of adding X_j to X_i's current parents. Thus, we evaluate each X_j by computing $\text{ScoreContribution}_{\text{BDe}}(X_i, \{X_j\} \cup \mathbf{Pa}^{n-1}(X_i) : D)$ where $\mathbf{Pa}^{n-1}(X_i)$ are the parents of X_i in the network found at the end of previous iteration. We then set C_i^n to consist of $\mathbf{Pa}^{n-1}(X_i)$ and the variables that maximize this score. We refer the reader to (Friedman, Nachman & Pe'er 1999) for more details on the algorithm and its complexity, as well as empirical results comparing its performance to traditional search techniques.

3.4 Discretization

In order to specify a Bayesian network model, we still need to define the local probability model for each variable. At the current stage, we choose to focus on the qualitative aspects of the data, and so we discretize gene expression values into three categories: $-1, 0$, and 1, depending on whether the expression level is significantly lower than, similar to, or greater than the respective control. The control expression level of a gene can be either determined experimentally (as in the methods of (DeRisi. et al. 1997)), or it can be set as the average expression level of the gene across experiments. The meaning of "significantly" is defined by setting a threshold to the ratio between measured expression and control. In our experiments we choose a threshold value of 0.5 in logarithmic (base 2) scale.

It is clear that by discretizing the measured expression levels we are loosing information. An alternative to discretization is using (semi)parametric density models for representing conditional probabilities in the networks we learn (e.g (Heckerman & Geiger 1995, Lauritzen & Wermuth 1989, Hoffman & Tresp 1996)). However, a bad choice of the parametric family can strongly bias the learning algorithm. We believe that discretization provides a reasonably unbiased approach for dealing with this type of data. We are currently exploring the appropriateness of several density models for this type of data.

4 Application to Cell Cycle Expression Patterns

We applied our approach to the data of Spellman et al. (1998), containing 76 gene expression measurements of the mRNA levels of 6177 *S. cerevisiae* ORFs. These experiments measure six time series under different cell cycle synchronization methods. Spellman et al. (1998) identified 800 genes whose expression varied over the different cell-cycle stages. Of these, 250 clustered into 8 distinct clusters based on the similarity of expression profiles. We learned networks whose variables were the expression level of each of these 800

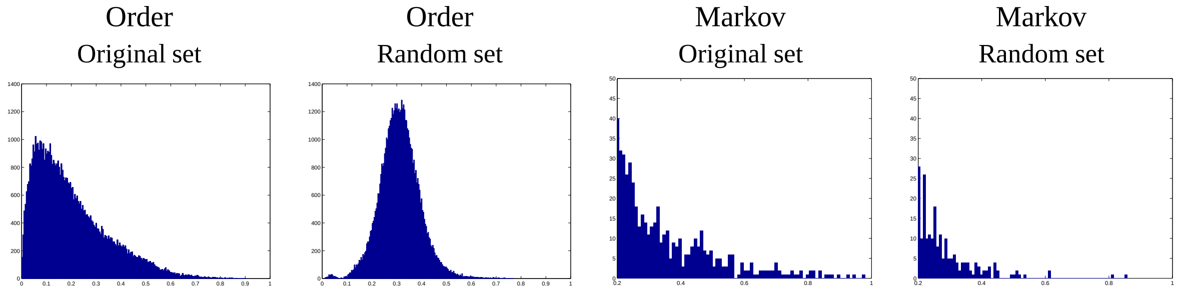

Figure 3: Histograms of the number of features at different confidence levels for the cell cycle data set, and the randomized data set. The histograms on the left are of order relations, and the on the right for Markov relations These histograms are all based on the 250 genes data set.

genes. Some of the robustness analysis was performed only on the set of 250 genes that appear in the 8 major clusters.

In learning from this data, we treat each measurement as a sample from a distribution, and do not take into account the temporal aspect of the measurement. Since it is clear that the cell cycle process is of temporal nature, we compensate by introducing an additional variable that denotes the cell cycle phase. This variable is then forced to be a root in all the networks learned. Its presence allows to model dependency of expression levels on current cell cycle.[2]

From this data set, we learned Bayesian networks using the Sparse Candidate algorithm with a 200-fold bootstrap. The learned features show that we can recover intricate structure even from such small data sets. It is important to note that our learning algorithm uses **no prior biological knowledge nor constraints**. All learned networks and relations are based solely on the information contained in the measurements themselves. These results are available at our WWW site:
`http://www.cs.huji.ac.il/labs/pmai2/expression`.

4.1 Robustness Analysis

We performed a number of tests to analyze the statistical significance and robustness of our procedure. We carried most of these tests on the smaller 250 gene data set for computational reasons.

To test the credibility of our confidence assessment, we created a random data set by randomly permuting the order of the experiments independently for each gene. Thus for each gene the order was random, but the composition of the series remained unchanged. In such a data set, genes are independent of each other, and thus we do not expect to find "real" features. As expected, both order and Markov relations in the random data set have significantly lower confidence. We compare the distribution of confidence estimates between the original data set and the randomized one in Figure 3. Clearly, the distribution of confidence estimates in the original data set have a longer and heavier tail in the high confidence region. Also, the confidence distribution for the real data set is concentrated closer to zero. This suggests that the networks learned from the real data are sparser.

Our analysis involves less than 15% of the *S. cerevisae* genes. This raises concern that omissions of genes lead to spurious conclusions. To estimate whether such conclusions appear in our analysis, we tested the robustness of our analysis to the addition of more genes, comparing the confidence of the learned features between the 250 and 800 gene data sets. We found a strong linear correlation between confidence levels of features obtained from the two data sets.

A crucial choice in our procedure is the threshold level used for discretization of the expression levels.

[2] We note that we can also learn temporal models using a Bayesian network that includes gene expression values in two (or more) consecutive time points (Friedman et al. 1998). We are currently perusing this issue.

Table 1: List of dominant genes in the ordering relations (top 14 out of 30)

Gene/ORF	Dominance	# of descendent genes		
	Score	$> .8$	$> .7$	notes
YLR183C	551	609	708	Contains forkheaded assosiated domain, thus possibly nuclear
MCD1	550	599	710	Mitotic chromosome determinant, null mutant is inviable
CLN2	497	495	654	Role in cell cycle START, null mutant exhibits G1 arrest
SRO4	463	405	639	Involved in cellular polarization during budding
RFA2	456	429	617	Involved in nucleotide excision repair, null mutant is inviable
YOL007C	444	367	624	
GAS1	433	382	586	Glycophospholipid surface protein, Null mutant is slow growing
YOX1	400	243	556	Homeodomain protein that binds leu-tRNA gene
YLR013W	398	309	531	
POL30	376	173	520	Required for DNA replication and repair, Null mutant is inviable
RSR1	352	140	461	GTP-binding protein of the ras family involved in bud site selection
CLN1	324	74	404	Role in cell cycle START, null mutant exhibits G1 arrest
YBR089W	298	29	333	
MSH6	284	7	325	Required for mismatch repair in mitosis and meiosis

It is clear that by setting a different threshold, we would get different discrete expression patterns. Thus, it is important to test the robustness and sensitivity of the high confidence features to the choice of this threshold. This was tested by repeating the experiments using different threshold levels. Again, the graphs show a definite linear tendency in the confidence estimates of features between the different discretization thresholds.

4.2 Biological Analysis

We believe that the results of this analysis can be indicative of biological phenomena in the data. This is confirmed by our ability to predict sensible relations between genes of known function. We now examine several consequences that we have learned from the data. We consider, in turn, the order relations and Markov relations found by our analysis. We only very briefly summarize a few of these.

4.2.1 Order Relations The most striking feature of the high confidence order relations, is the existence of *dominant genes*. Out of all 800 genes only few seem to dominate the order (i.e., appear before many genes). The intuition is that these genes are indicative of potential causal sources of the cell-cycle process. Let $C_o(X, Y)$ denote the confidence in X being ancestor of Y. We define the *dominance score* of X as $\sum_{Y, C_o(X,Y)>t} C_o(X, Y)^k$, using the constant k for rewarding high confidence features and the threshold t to discard low confidence ones. We refer to genes with high dominance score as dominant genes. These genes are extremely robust to parameter selection for both t, k and the discretization cutoff of Section 3.4. A list of the highest scoring dominant genes appears in Table 1.

Inspection of the list of dominant genes reveals quite a few interesting features. Among the dominant genes are those directly involved in cell-cycle control and initiation (e.g., CLN1, CLN2 and CDC5) and genes whose null mutant is inviable (e.g., MCD1 and RFA2). These are clearly key genes in basic cell functions. Most of the dominant genes are nuclear proteins, and some of the unknown genes are also potentially nuclear: (e.g., YLR183C contains a forkhead-associated domain which is found almost entirely among nuclear proteins). Many of these genes are components of pre-replication complexes and involve very early steps of replication. Such functions are prior conditions to most processes in the nucleus and the cell in general. A few non nuclear dominant genes are localized in the cytoplasm membrane (SRO4,GAS1 and RSR1). These are involved in the budding and sporulation process which have an important role in the cell-cycle.

We also note that many of the *dominated* genes (i.e. are caused by dominators with high confidence) are

Table 2: List of top Markov relations

Confidence	Gene 1	Gene 2	notes
1.0	YKL163W-PIR3	YKL164C-PIR1	Close locality on chromosome
0.985	PRY2	YKR012C	No homolog found
0.985	MCD1	MSH6	Both bind to DNA during mitosis
0.98	PHO11	PHO12	Both nearly identical acid phosphatases
0.975	HHT1	HTB1	Both are Histones
0.97	HTB2	HTA1	Both are Histones
0.94	YNL057W	YNL058C	Close locality on chromosome
0.94	YHR143W	CTS1	Homolog to EGT2 cell wall control, both do cytokinesis
0.92	YOR263C	YOR264W	Close locality on chromosome
0.91	YGR086	SIC1	
0.9	FAR1	ASH1	Both part of a mating type switch, **expression uncorelated**
0.89	CLN2	SVS1	Function of SVS1 unknown, possible regulation mediated through SWI6
0.88	YDR033W	NCE2	Homolog to transmembrame proteins, suggesting both involved in protein secretion
0.86	STE2	MFA2	A mating factor and receptor
0.85	HHF1	HHF2	Both are Histones
0.85	MET10	ECM17	Both are sulfite reductases
0.85	CDC9	RAD27	Both participate in Okazaki fragment processing

themselves part of the replication machinery, (e.g. CDC54 and MCM2), or are transcription regulators (e.g. RME1,ASH1, and TEC1). These causal relations do not only make sense but also show that high confidence order relations identify pairs of genes which are close (i.e. with small number of intermediate factors) in the causal pathway.

4.2.2 Markov Relations Inspection of the top Markov relations reveals that most pairs are functionaly related. A list of the top scoring relations can be found in Table 2. Among these, all involving two known genes (10/20) make sense biologically. When one of the ORFs is unknown careful searches using Psi-Blast (Altschul et al. 1997), Pfam (Sonnhammer et al. 1998) and Protomap (Yona et al. 1998) can reveal firm homologies to proteins functionally related to the other gene in the pair. (e.g. YHR143W, which is paired to the endochitinase CTS1, is related to EGT2 - a cell wall maintenance protein). Several of the unknown pairs are physically adjacent on the chromosome, and thus presumably regulated by the same mechanism. Such analysis raises the number of biologically sensible pairs to 17/20. For the other 3 pairs no clear homology could be assigned.

There are some interesting Markov relations found that are beyond the limitations of clustering techniques. One such regulatory link is FAR1-ASH1: both proteins are known to participate in a mating type switch. The correlation of their expression patterns is low and (Spellman et al. 1998) cluster them into different clusters. Among the high confidence markov relations, one can also find examples of conditional indpendence, i.e., a group of highly correlated genes whose correlation can be explained within our network stucture. One such example involves the genes: CLN2,RNR3,SVS1,SRO4 and RAD41, their expression is correlated, in (Spellman et al. 1998) all appear in the same cluster. In our network CLN2 is with high confidence a parent of each of the other 4 genes, while no links are found between them. This suits biological knowledge: CLN2 is a central cell cycle control while there is no clear biological relationship between the others.

5 Discussion and Future Work

In this paper we presented a new approach for analyzing gene expression data that builds on theory and algorithms for learning Bayesian networks. We described how one can apply these techniques to gene expression data. The approach includes two techniques that were motivated by the challanges posed by this domain: a novel search algorithm (Friedman, Nachman & Pe'er 1999) and an approach for estimating statistical confidence (Friedman, Goldszmidt & Wyner 1999). We applied our methods to the real expression data of Spellman et al. (1998). Although we did not use any prior knowledge, we managed to extract many biologically plausible conclusions from this analysis.

Our approach is quite different than the clustering approach used by (Ben-Dor & Yakhini 1999, Alon et al. 1999, Eisen et al. 1998, Michaels et al. 1998, Spellman et al. 1998), in that it attempts to learn a much richer structure from the data. Our methods are capable of discovering causal relationships, interactions between genes other than positive correlation, and finer intra-cluster structure. We are currently developing hybrid approaches that combine our methods with clustering algorithms to learn models over "clustered" genes.

The biological motivation of our approach is similar to work on inducing *genetic networks* from data (Akutsu et al. 1998, Chen et al. 1999, Somogyi et al. 1996, Weaver et al. 1999). There are two key differences: First, the models we learn have probablistic semantics. This better fits the stochastic nature of both the biological processes and noisy experimentation. Second, our focus is on extracting features that are pronounced in the data, in contrast to current genetic network approaches that attempt to find a single model that explains the data.

We are currently working on improving methods for expression analysis by expanding the framework described in this work. Promising directions for such extentions are: (a) Developing the theory for learning local probability models that are capable of dealing with the continuous nature of the data; (b) Improving the theory and algorithms for estimating confidence levels; (c) Incorporating biological knowledge (such as possible regulatory regions) as prior knowledge to the analysis; (d) Improving our search heuristics; (e) Applying *Dynamic Bayesian Networks* (Friedman et al. 1998) to temporal expression data.

Finally, one of the most exciting longer term prospects of this line of research is discovering causal patterns from gene expression data. We plan to build on and extend the theory for learning causal relations from data and apply it to our domain. The theory of causal networks allows learning both from observational data and *interventional* data, where the experiment intervenes with some causal mechanisms of the observed system. In the context of gene expression, we should view knockout/overexpressed mutants as such interventions. Thus, we can design methods that deal with mixed forms of data in a principled manner (See (Cooper & Yoo 1999) for a recent work in this direction). In addition, this theory can provide tools for *experimental design*, that is, understanding which interventions are deemed most informative to determining the causal structure in the underlying system.

Acknowledgements The authors are grateful to Hadar Benyaminy, David Engelberg, Moises Goldszmidt, Daphne Koller, and especially Itzik Pe'er for comments on drafts of this paper and useful discussions relating to this work. This work was supported through the generosity of the Michael Sacher Trust.

References

Akutsu, S., Kuhara, T., Maruyama, O. & Minyano, S. (1998), Indentification of gene regulatory networks by strategic gene disruptions and gene over-expressions, *in* 'SODA', ACM-SIAM.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D. & Levine, A. J. (1999), 'Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays', *Proc. Nat. Acad. Sci. USA* **96**, 6745–6750.

Altschul, S., Thomas, L., Schaffer, A., Zhang, J. Zhang, Z., Miller, W. & Lipman, D. (1997), 'Gapped blast and psi-blast: a new generation of protein database search programs', *Nucleic Acids Res* **25**.

Barron, A. R. & Cover, T. M. (1991), ‘Minimum complexity density estimation’, *IEEE Trans. on Info. Theory* **37**.

Ben-Dor, A. & Yakhini, Z. (1999), Clustering gene expression patterns, *in* ‘RECOMB’.

Besnard, P. & Hanks, S., eds (1995), *Proc. Eleventh Conference on Uncertainty in Artificial Intelligence (UAI ’95)*, Morgan Kaufmann, San Francisco, Calif.

Chen, T., Filkov, V. & Skiena, S. (1999), dentifying gene regulatory networks from experimental data, *in* ‘RECOMB’.

Chickering, D. M. (1995), A transformational characterization of equivalent Bayesian network structures, *in* Besnard & Hanks (1995), pp. 87–98.

Chickering, D. M. (1996), Learning Bayesian networks is NP-complete, *in* D. Fisher & H.-J. Lenz, eds, ‘Learning from Data: Artificial Intelligence and Statistics V’, Springer Verlag.

Cooper, G. F. & Herskovits, E. (1992), ‘A Bayesian method for the induction of probabilistic networks from data’, *Machine Learning* **9**, 309–347.

Cooper, G. F. & Moral, S., eds (1998), *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI ’98)*, Morgan Kaufmann, San Francisco, Calif.

Cooper, G. & Yoo, C. (1999), Causal discovery from a mixture of experimental and observational data, *in* Dubios & Laskey (1999).

DeRisi., J., Iyer, V. & Brown, P. (1997), ‘Exploring the metabolic and genetic control of gene expression on a genomic scale’, *Science* **282**, 699–705.

Dubios, H. & Laskey, K., eds (1999), *Proc. Fifthteenth Conference on Uncertainty in Artificial Intelligence (UAI ’99)*, Morgan Kaufmann, San Francisco, Calif.

Efron, B. & Tibshirani, R. J. (1993), *An Introduction to the Bootstrap*, Chapman & Hall, London.

Eisen, M., Spellman, P., Brown, P. & Botstein, D. (1998), ‘Cluster analysis and display of genome-wide expression patterns’, *Proc. Nat. Acad. Sci. USA* **95**, 14863–14868.

Friedman, N., Geiger, D. & Goldszmidt, M. (1997), ‘Bayesian network classifiers’, *Machine Learning* **29**, 131–163.

Friedman, N., Goldszmidt, M. & Wyner, A. (1999), Data analysis with bayesian networks: A bootstrap approach, *in* Dubios & Laskey (1999).

Friedman, N., Murphy, K. & Russell, S. (1998), Learning the structure of dynamic probabilistic networks, *in* Cooper & Moral (1998), pp. 139–147.

Friedman, N., Nachman, I. & Pe’er, D. (1999), Learning bayesian network structure from massive datasets: The “sparse candidate” algorithm, *in* Dubios & Laskey (1999).

Friedman, N. & Yakhini, Z. (1996), On the sample complexity of learning Bayesian networks, *in* E. Horvitz & F. Jensen, eds, ‘Proc. Twelfth Conference on Uncertainty in Artificial Intelligence (UAI ’96)’, Morgan Kaufmann, San Francisco, Calif.

Heckerman, D. & Geiger, D. (1995), Learning Bayesian networks: a unification for discrete and Gaussian domains, *in* Besnard & Hanks (1995), pp. 274–284.

Heckerman, D., Geiger, D. & Chickering, D. M. (1995), 'Learning Bayesian networks: The combination of knowledge and statistical data', *Machine Learning* **20**, 197–243.

Heckerman, D., Meek, C. & Cooper, G. (1997), A bayesian approach to causal discovery, Technical report. Technical Report MSR-TR-97-05, Microsoft Research.

Höffgen, K. L. (1993), Learning and robust learning of product distributions, *in* 'COLT '93', pp. 77–83.

Hoffman, R. & Tresp, V. (1996), Discovering structure in continuous variables using bayesian networks, *in* 'Advances in Neural Information Processing Systems 8 (NIPS '96)', MIT Press.

Iyer, V., Eisen, M., Ross, D., Schuler, G., Moore, T., Lee, J., Trent, J., Staudt, L., Hudson, J., Boguski, M., Lashkari, D., Shalon, D., Botstein, D. & Brown, P. (1999), 'The transcriptional program in the response of human fibroblasts to serum', *Science* **283**, 83–87.

Jensen, F. V. (1996), *An introduction to Bayesian Networks*, University College London Press, London.

Lauritzen, S. L. & Wermuth, N. (1989), 'Graphical models for associations between variables, some of which are qualitative and some quantitative', *Annals of Statistics* **17**, 31–57.

Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Want, C., Kobayashi, M., Horton, H. & Brown, E. L. (1996), 'DNA expression monitoring by hybridization of high density oligonucleotide arrays', *Nature Biotechnology* **14**, 1675–1680.

Michaels, G., Carr, D., Askenazi, M., Fuhrman, S., Wen, X. & Somogyi, R. (1998), Cluster analysis and data visualization for large scale gene expression data, *in* 'Pac. Symp. Biocomputing', pp. 42–53.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Francisco, Calif.

Pearl, J. & Verma, T. S. (1991), A theory of inferred causation, *in* J. A. Allen, R. Fikes & E. Sandewall, eds, 'Principles of Knowledge Representation and Reasoning: Proc. Second International Conference (KR '91)', Morgan Kaufmann, San Francisco, Calif., pp. 441–452.

Somogyi, R., Fuhrman, S., Askenazi, M. & Wuensche, A. (1996), The gene expression matrix: Towards the extraction of genetic network architectures, *in* 'WCNA96'.

Sonnhammer, E. L., Eddy, S., Birney, E., Bateman, A. & Durbin, R. (1998), 'Pfam: multiple sequence alignments and hmm-profiles of protein domains', *Nucl. Acids Res.* **26**, 320–322. http://pfam.wustl.edu/.

Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D. & Futcher (1998), 'Comprehensive identification of cell cycle-regulated genes of the yeast *sacccharomyces cerevisiae* by microarray hybridization', *Molecular Biology of the Cell* **9**, 3273–3297.

Spirtes, P., Glymour, C. & Scheines, R. (1993), *Causation, prediction, and search*, Springer-Verlag.

Thiesson, B., Meek, C., Chickering, D. M. & Heckerman, D. (1998), Learning mixtures of Bayesian networks, *in* Cooper & Moral (1998).

Weaver, D., Workman, C. & Stormo, G. (1999), Modeling regulatory networks with weight matrices, *in* 'Pac. Symp. Biocomputing', pp. 112–123.

Wen, X., Furhmann, S., Micheals, G. S., Carr, D. B., Smith, S., Barker, J. L. & Somogyi, R. (1998), 'Large-scale temporal gene expression mapping of central nervous system development', *Proc. Nat. Acad. Sci. USA* **95**, 334–339.

Yona, G., Linial, N., Tishby, N. & Linial, M. (1998), A map of the protein space - an automatic hierarchical classification of all protein sequences, *in* ‘ISMB’. http://protomap.cs.huji.ac.il.