Analysis of Gene Expression Data **Spring Semester, 2002**

Lecture 10: June 20, 2002

Lecturer: Nir Friedman *Scribe: Dina Dushnik and Adar Shtainhart*

10.1 Introduction

10.1.1 Motivation

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, yet their protein makeup can be drastically different both temporally and spatially, due to regulation. 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 order to get a "snapshot" of transcription levels of the genes within the cell we use *DNA microarrays* which measure the expression level for thousands of genes (the data matrix). An analysis of this data will give us some understanding on the key biological features of the cellular systems that produced it.

One analysis approach is using *clustering* algorithms which attempt to locate group of genes that have similar expression patterns over a set of experiments (for more details see lecture notes 4-6). This approach has proven to be useful in discovering genes that are co-regulated. A more ambitious goal for analysis is revealing the structure of the transcriptional regulation process. This is clearly a hard problem. The current data is extremely noisy. Moreover, mRNA expression data alone only gives a partial picture that does not reflect key events such as translation and protein (in)activation. Finally, 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.

In this lecture, we describe another approach for analyzing gene expression patterns (based on the work of Friedman et al. [12]), that uncovers properties of the transcriptional program by examining statistical properties of *dependence* and *conditional independence* in the data based on *Bayesian networks* [24]. These networks represent the dependence structure between multiple interacting quantities by a graph-based model. Such models are attractive for their ability to describe complex stochastic processes, and for providing clear methodologies for learning from (noisy) observations.

10.1.2 Advantages of Using Bayesian Networks

- Bayesian networks 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.

- The statistical foundations for learning Bayesian networks from observations, and computational algorithms to do so are well understood and have been used successfully in many applications.

- Bayesian networks provide models of causal influence as will be discussed later.

10.2 Bayesian Networks

10.2.1 Representing Distributions with Bayesian Networks

Suppose we are given a set of assertions and a variety of ways in which they support each other. Each assertion establishes a value for an attribute and is of the form $(X_i = x_i)$, that is, "Variable X_i has value x_i". The variables are $X_1, \ldots, X_n$. We would know everything we need to know about the world described by these assertions if we had the joint probability $P(X_1, \ldots, X_n)$. From this probability function we could compute any other probability such as $P(X2)$ or $P(X2|X3, X5)$. Unfortunately, assuming for simplicity that the variables are binary, the representation complexity of $P(X_1, \ldots, X_n)$ is 2^n, which is impractical even for small value of n.

Bayesian Networks simplify this problem by taking advantage of existing causal connections between assertions, and of assumptions about conditional independence. 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* (DAG) 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.

Definition We say that x is **conditional independent** of y given z if $P(x, y|z) = p(x|z)p(y|z)$ or alternatively $p(x|y, z) = p(x|z)$.

By applying the chain rule of probabilities and properties of conditional independencies, any joint distribution that satisfies the markov assumption can be decomposed in the *product* *form*

$$P(X_1,\ldots,X_n)=\prod_{i=1}^{n}P(X_i|\mathbf{Pa}(X_i)), \tag{10.1}$$

In Figure 10.1 we can see an example of a simple Bayesian network structure. This network describes the connections between the following events:

- B - There is a burglary.
- A - The alarm goes off.
- E - There is an earthquake.
- R - There is a radio report of an earthquake.
- C - Mr. Watson reports hearing the alarm.

where the list of independences is:

- $I(E,B)$ - Given the parents of $E(\emptyset)$, E is conditionally independent of B.
- $I(B,(E,R))$ - Given the parents of $B(\emptyset)$, B is conditionally independent of E and R.
- $I(R,(B,A,C)|E)$ - Given the parents of $R(E)$, R is conditionally independent of B, A and C.
- $I(A,R|(B,E))$ - Given the parents of $A(B,E)$, A is conditionally independent of R.
- $I(C,(R,B,E)|A)$ - Given the parents of $C(A)$, C is conditionally independent of R, B and E.

According to equation 10.1, given the graph the joint distribution of the five events is:

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

We get a representation complexity of $O(n2^{k+1})$ where n is number of variables and k is the maximum number of possible parents.

To fully specify a joint distribution, we also need to specify the conditional probabilities in the product form. The second part of the Bayesian network describes these conditional distributions, $P(X_i|\mathbf{pa}(X_i))$ for each variable X_i. We denote the parameters that specify these distribution by Θ. In general, we can choose the form of the conditional distribution:

- Discrete variables - In the case of finite valued variables, we can represent these conditional distributions as tables (see Figure 10.2 for an example).

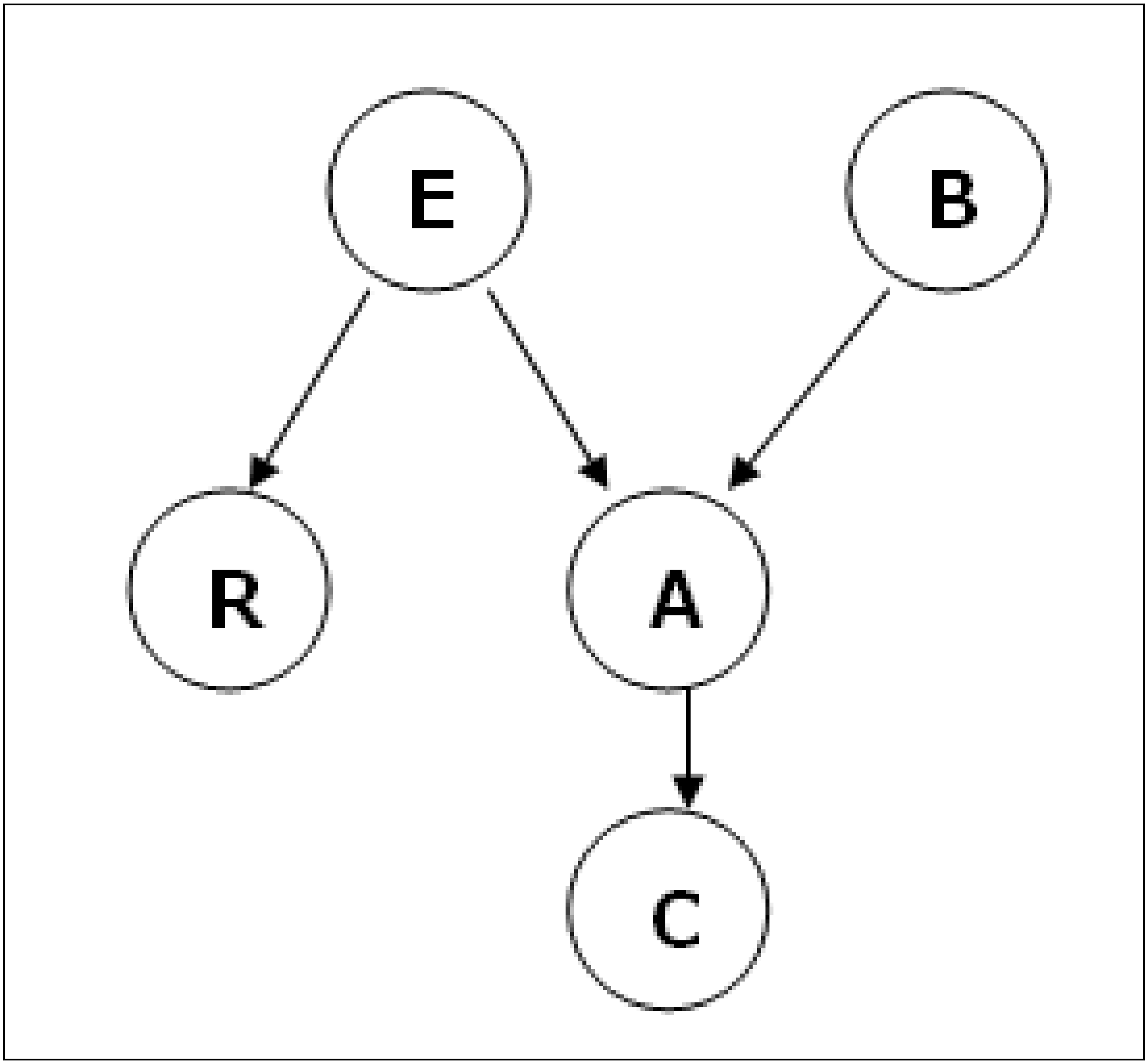
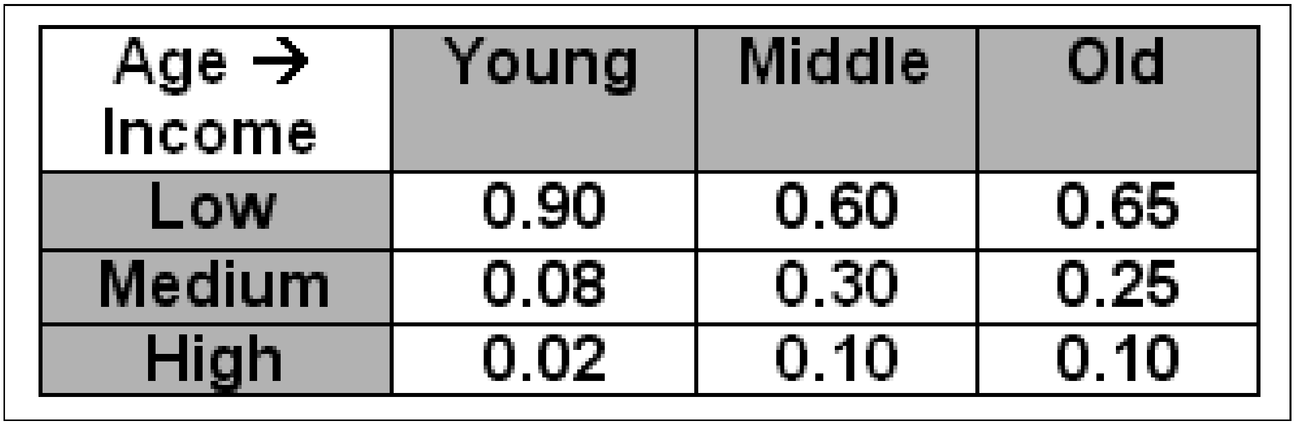


Figure 10.1: A simple Bayesian network structure.

- Continuous variables - Unlike the case of discrete variables, when the variable and its parents are real valued, there is no representation of all possible densities. We can use *linear Gaussians* conditional densities in order to represent multivariate continuous distributions. That is, the variable is normally distributed around a mean that depends *linearly* on the value of its parents. The variance of this normal distribution is independent of the parents' values. If all the variables in the network have linear Gaussian conditional distributions, then the joint distribution is a multivariate Gaussian [22].

- Hybrid networks - When the networks contains continuous variables with discrete parents we can use *conditional gaussian distributions* [22]. The case of a discrete value with continuous parents is not allowed.



Age → Income	Young	Middle	Old
Low	0.90	0.60	0.65
Medium	0.08	0.30	0.25
High	0.02	0.10	0.10

Figure 10.2: Source: [35]. An example for discrete variables. This table contains the values of $P(B|A)$, where A is Age, with values (Young, Middle, Old) and B is Income, with values (Low, Medium, High).

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 exploiting the explicit representation of structure in order to answer queries efficiently (see [24, 21]).

Following are some examples for Bayesian networks usage:

- *Linkage Analysis* [31] - "Linkage" refers to the tendency of certain genes to be inherited together. Two genes are said to be "linked" if they are often inherited together, due to their close proximity on a chromosome. Linkage analysis is a tool that enables us to describe a family genotype tree and to find from which parent a specifics gene has been inherited. We can learn about a child given information about its parents (no information about the grandparent is required).

- *Phylogenetic tree* [30] - evolutionary hypotheses represented as a dendrogram or branching diagram.
- *Markov chain* [32] - A collection of random variables having the property that, given the present, the future is conditionally independent of the past.

10.2.2 Representing Equivalence Classes of Bayesian Networks

Let $\text{Ind}(G)$ be a set of independence statements (of the form X is independent of Y given Z). 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., $\text{Ind}(G) = \emptyset$). We say that two graphs G and G' are *equivalent* if $\text{Ind}(G) = \text{Ind}(G')$.

Definition A *v-structure* in a graph is a structure of two directed edges converging into the same node, such as $a \rightarrow b \leftarrow c$.

Theorem 10.1 *[25] Two DAGs are equivalent if and only if they have the same underlying undirected graph and the same v-structures.*

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 arc $X \rightarrow Y$; an undirected edge X—Y denotes that some members of the class contain the arc $X \rightarrow Y$, while others contain the arc $Y \rightarrow X$. Given a DAG G, the PDAG representation of its equivalence class can be constructed efficiently [3].

10.2.3 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. More precisely, we search for an equivalence class of networks that best matches D. In the following we outline the learning method. For more information on the subject see [17]. 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. One method for deriving a score is based on Bayesian considerations (see [5, 19]). In this score, we evaluate the posterior probability of a graph given the data:

$$\begin{aligned} \mathrm{S}(G : D) &= \log P(G \mid D) \\ &= \log P(D \mid G) + \log P(G) + C \end{aligned}$$

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 [15]. 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. We will use the priors described by Heckerman and Geiger for hybrid networks of multinomial distributions and conditional Gaussian distributions (see [17, 18]). Assuming that the data set is a *complete data* (a data set in which each instance contains the value of all the variables in the network) several properties are satisfied by these priors:

- The priors are *structure equivalent*, i.e., if G and G' are equivalent structures they are guaranteed to have the same score.
- The priors are *decomposable*. That is, the score can be rewritten as the sum

  $$Score(G : D) = \sum_i ScoreContribution(X_i, \mathbf{Pa}(X_i) : D),$$

  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.
- The local contributions for each variable can be computed using a closed form equation [18].

Finding the structure G that maximizes the score is known to be NP-hard [4], thus we resort to heuristic search. A *local* search procedure that changes one arc at each move can efficiently evaluate the gains made by adding, removing or reversing a single arc. 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. Examples of other search methods that advance using one-arc changes include beam-search, stochastic hill-climbing, and simulated annealing.

10.2.4 Learning Causal Patterns

In order to analyze the mechanism that generated the dependencies 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. A causal network 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 is that this model views the parents of a variable as its *immediate causes*. For example assume X is a transcription factor of Y, so there is an edge $X \to Y$. If we knockout gene X this will affect the expression of gene Y, but a knockout of gene Y has no effect on the expression of gene X. We can relate causal networks and Bayesian networks, by assuming the *Causal Markov Assumption*. When this assumption holds, the causal network satisfies the Markov independencies of the corresponding Bayesian network, thus allowing us to treat causal networks as Bayesian networks.

The central issue is: when can we learn a causal network from *observation* (a passive measurement of our domain)? (see [20, 25, 28]). From observations alone, we cannot distinguish between causal networks that specify the same independence assumptions, i.e., belong to the same equivalence class. Thus, if we are willing to accept the causal markov assumption and we can learn a PDAG from the data, then we can recover some of the causal directions. Moreover, by using Theorem 10.1, we can predict what aspects of the proposed model can be recovered based on observations alone. The situation is more complex when we have a combination of observations and results of different interventions. From such data we might be able to distinguish between equivalent structures [6].

10.3 Applying Bayesian Networks to Expression Data

This section describes an approach for 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 depend 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 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 the data.

Most of the difficulties in learning from expression data revolve around the following central point: Contrary to previous applications of learning Bayesian networks, expression data involves transcript levels of thousands of genes while current datasets 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.

10.3.1 Network Features

When learning models with many variables, small datasets are not sufficiently informative to significantly determine that a single model is the "right" one. Instead, many different networks should be considered as reasonable explanation of the given the data. From a Bayesian perspective, we say that the posterior probability over models is not dominated by a single model (or equivalence class of models). We would like to analyze this set of plausible (i.e., high-scoring) 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 discuss two classes of features involving pairs of variables:

Markov Relations

A relation of this type specifies if Y is in the *Markov blanket* of X, where the Markov blanket of X is the minimal set of variables that *shield* X from the rest of the variables in the model (see Figure 10.3 for an example). More precisely, X 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 [24]. 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, that 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 in their interaction.

Order Relations

An order relation specifies if X is an ancestor of Y in all the networks of a given equivalence class. That is, if the given PDAG contains a directed path from X to Y. This type of relation does not involve only a close neighborhood, but rather captures a global property. Recall that under the assumptions of Section 10.2.4,

learning that X is an ancestor of Y 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, rather than evidence, that X might be a causal ancestor of Y.

While at this point we handle only pairwise features, it is clear that this analysis is not restricted to them, and we should examine also features that are more complex (see [26]).

10.3.2 Estimating Statistical Confidence in Features

We now face the following problem: To what extent do 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) (see Figure 10.5).

An effective, and relatively simple, approach for estimating confidence is the *bootstrap* method [9]. The main idea behind the bootstrap is simple. We generate "perturbed" versions of the original dataset, 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 effect many of the features.

In our context, we use the bootstrap as follows:

- For $i = 1 \ldots m$ (in the experiments, we set $m = 200$):
  - Resample with replacement N instances from D. Denote by D_i the resulting dataset.
  - Apply the learning procedure on D_i to deduce a network structure $\hat{G}_i$.
- For each feature f of interest calculate

  $$\mathrm{conf}(f) = \frac{1}{m} \sum_{i=1}^{m} f(\hat{G}_i)$$

  where $f(G)$ is 1 if f is a feature in G, and 0 otherwise.

10.3.3 Efficient Learning Algorithms

In section 10.2.3 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.

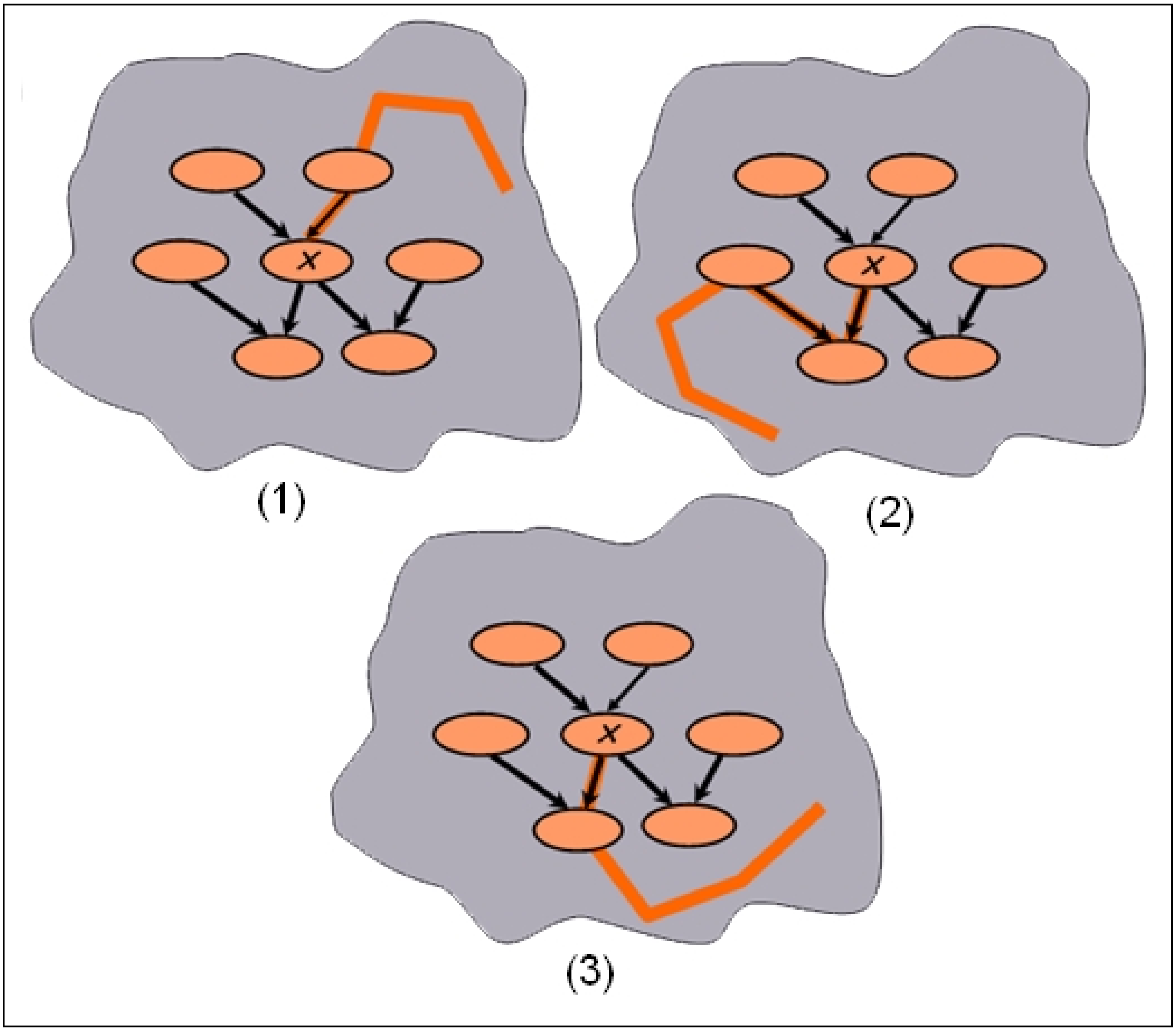


Figure 10.3: Sorce [34]. An example for a Markov blanket. This Markov blanket of X contains all paths from X to other nodes. There are three kinds of such paths as shown in the figure: (1) *Upward paths* the parents of X. (2) *Sideway paths* the spouses of X. (3) *Downward paths* the children of X.

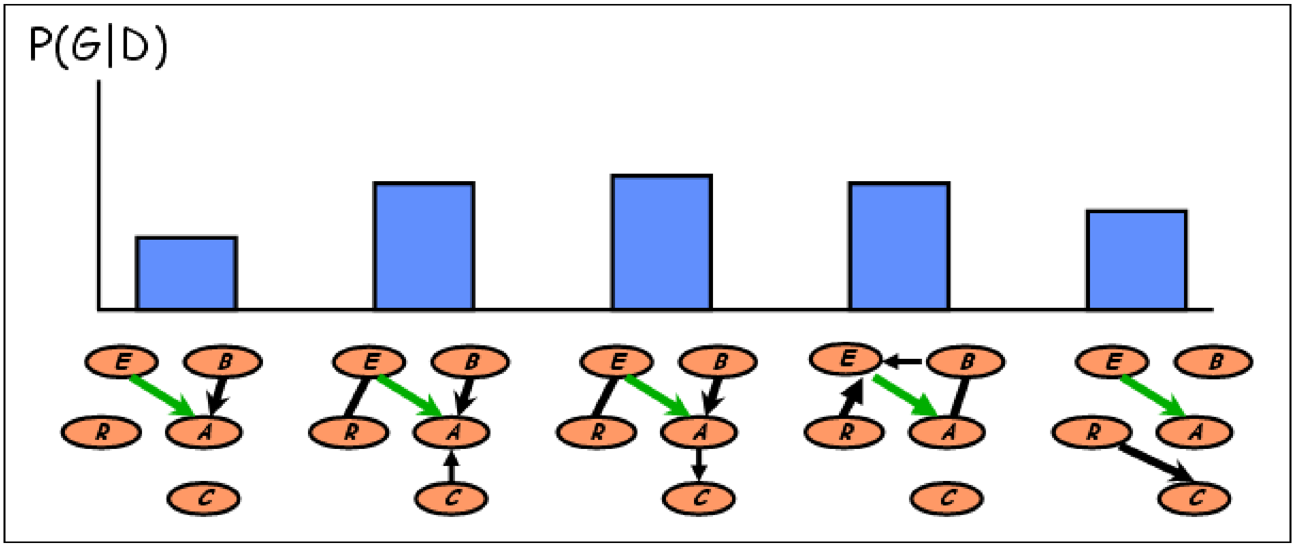


Figure 10.4: An example of high scoring networks that have a common feature.

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 [14]. 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.

10.4 Experimental Results

The Byesian Networks approach was applied by Freidman et al. [12] to two datasets: the data of Spellman et al. [10] and the data of Hughes et al. [11]. From this point and on, we will refer only to the data of Spellman et al. We refer the reader to [26] for details about the results from the data of Hughes et al.

The data contains 79 gene expression measurements of the mRNA levels of 6177 *S. cerevisiae* ORFs. These experiments measure expression in fixed time intervals under different cell cycle synchronization methods. Spellman et al. identified 800 genes whose expression varied over the different cell-cycle stages. They clustered these 800 genes, based on the similarity of expression profiles, resulting 8 major clusters, which contained 250 genes in total. The variables of the learned networks were the expression level of each of these 800 genes. Some of the robustness analysis was performed only on the set of 250 genes that appear in the 8 major clusters.

Freidman et al. [12] used the Sparse Candidate algorithm with a 200-fold bootstrap in the learning process. The learned features show that intricate structure can be recovered even from such small data sets. It is important to note that this learning algorithm uses no prior biological knowledge nor constraints. All learned networks and relations are based solely on the information conveyed in the measurements themselves. These results are available in [33]. Figure 10.5 illustrates the graphical display of results of this analysis.

10.4.1 Robustness Analysis

Freidman et al. performed a number of tests to analyze the statistical significance and robustness of their procedure. They carried most of these tests on the smaller 250 gene data set for computational reasons.

To test the credibility of their confidence assessment, they 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. Clearly, the distribution of confidence estimates in the original data set have a longer and heavier tail in the high confidence region. The runs on the random data sets do not learn almost anything with a confidence level above 0.8, which can lead us to believe that most features that are learned in the original data set with such confidence levels originate in true signals in the data. Also, the confidence distribution for the real dataset is concentrated closer to zero than the random distribution. This suggests that the networks learned from the real data are sparser.

Since the analysis was not performed on the whole *S. cerevisiae* genome, Freidman et al. also tested the robustness of their analysis to the addition of more genes, comparing the confidence of the learned features between the 250 and 800 gene datasets.

10.4.2 Biological Analysis

Freidman et al. believe that the results of this analysis can be indicative of biological phenomena in the data. This is confirmed by their ability to predict sensible relations between genes of known function. We now examine several consequences from this analysis. We consider, in turn, the order relations and Markov relations found.

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

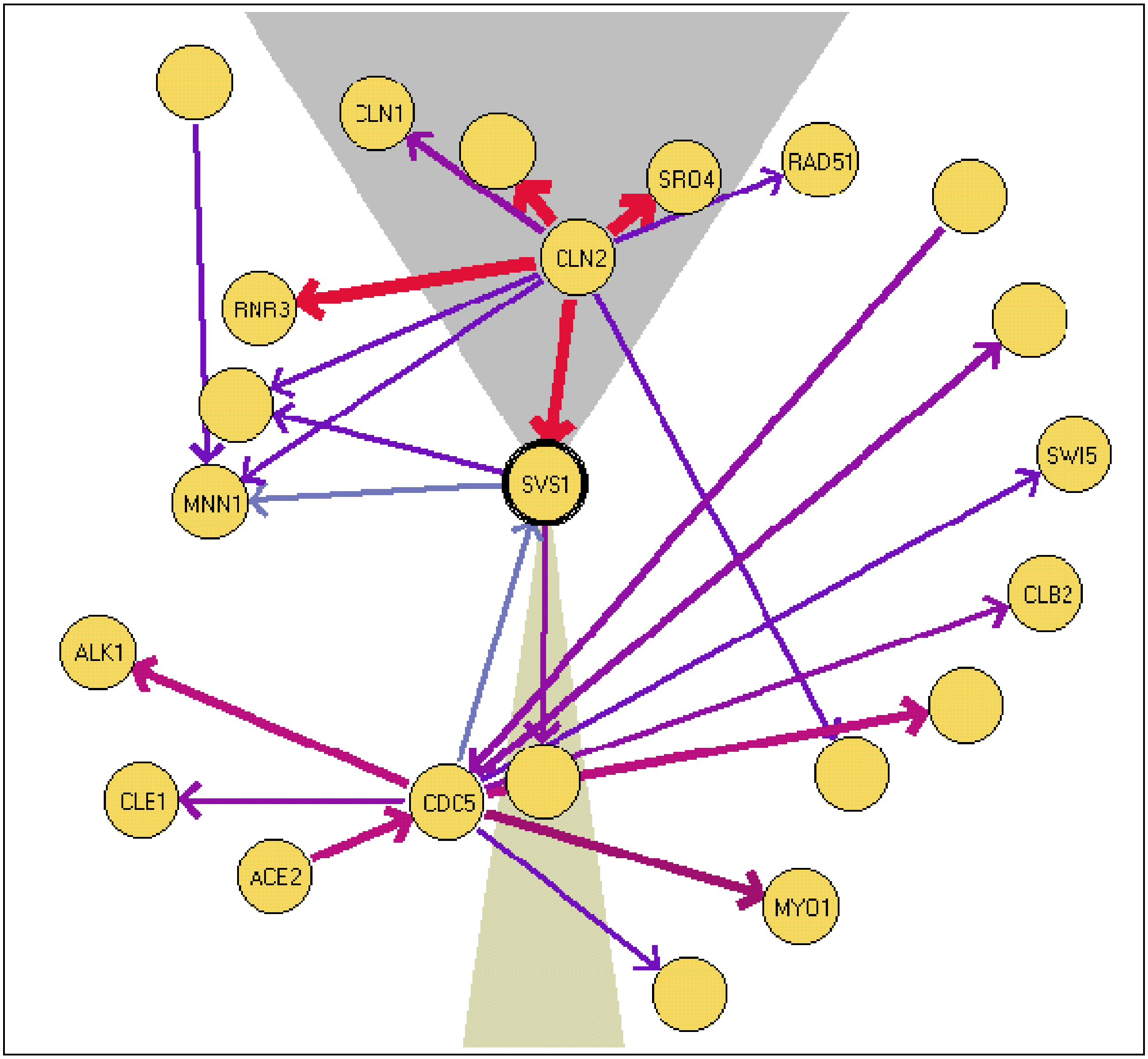


Figure 10.5: Source: [12]. An example of the graphical display of Markov features. This graph shows a ”local map” for the gene SVS1. The width (and color) of edges corresponds to the computed confidence level. An edge is directed if there is a sufficiently high confidence in the order between the pair genes connected by the edge. This local map shows that CLN2 separates SVS1 from several other genes. Although there is a strong connection between CLN2 to all these genes, there are no other edges connecting them. This indicates that, with high confidence, these genes are conditionally independent given the expression level of CLN2.

genes). The intuition is that these genes are indicative of potential causal sources of the cell-cycle process. A list of the highest scoring dominating genes appears in Table 10.1.

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. For example, CLN1, CLN2 and CDC5, whose functional relation has been established [7, 8]. Other genes, like MCD1 and RFA2, were found to be essential [16]. These are clearly key genes in basic cell functions, involved in chromosome dynamics and stability (MCD1) and in nucleotide excision repair (RFA2). Most of the dominant genes encode 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). Some of them are components of pre-replication complexes. Others (like RFA2,POL30 and MSH6) are involved in DNA repair. It is known that DNA repair is a prerequisite for transcription, and DNA areas which are more active in transcription, are also repaired more frequently [23, 29].

A few non nuclear dominant genes are localized in the cytoplasm membrane (SRO4 and RSR1). These are involved in the budding and sporulation process which have an important role in the cell-cycle. RSR1 belongs to the ras family of proteins, which are known as initiators of signal transduction cascades in the cell.

Markov Relations

Inspection of the top Markov relations reveals that most are functionally related. A list of the top scoring relations can be found in Table 10.2. Among these, all involving two known genes make sense biologically. When one of the ORFs is unknown careful searches using Psi-Blast [1], Pfam [27] and Protomap [36] 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, are presumably regulated by the same mechanism (see [2]), although special care should be taken for pairs whose chromosomal location overlap on complementary strands, since in these cases we might see an artifact resulting from cross-hybridization.

There are some interesting Markov relations found that are not discovered using 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 [10] cluster them into different clusters. Among the high confidence markov relations, one can also find examples of conditional independence, i.e., a group of highly correlated genes whose correlation can be explained within the resulted network structure. One such example involves the genes: CLN2, RNR3, SVS1, SRO4 and RAD41. Their expression is correlated and in [10] all appear in the same cluster. In the resulting 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 and early cell cycle control, while there is no clear biological relationship between the others.

10.5 Improving The Framework

The framework we described here can be expanded in a number of promising directions:

- Developing the theory for learning local probability models that are capable of dealing with the continuous nature of the data.
- Improving the theory and algorithms for estimating confidence levels.
- Incorporating biological knowledge (such as possible regulatory regions) as prior knowledge to the analysis.
- Improving the search heuristics.
- Applying *Dynamic Bayesian Networks* ([13]) 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 can build on and extend the theory for learning causal relations from data and apply it to gene expression. 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 gene expression context, we can model knockout/overexpressed mutants as such interventions. Thus, we can design methods that deal with mixed forms of data in a principled manner (See [5] 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. Friedman et al. have extended their framework in this direction (see [26]).

Gene/ORF	Dominance	# of descendent genes		
	Score	> .8	> .7	Notes
YLR183C	551	609	708	Contains forkheaded associated 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	
YOX1	400	243	556	Homeodomain protein
GAT3	398	309	531	Putative GATA zinc finger transcription factor related to polII transcription
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

Table 10.1: Source [12]. List of dominant genes in the ordering relations (top 14 out of 30). The first column specifies the name of the gene/ORF, the second column specifies the level of dominance score of the gene/ORF as appeared in the experiments results, the next column contains the number of descendent genes with a level of confidence higher than 0.8, the next column contains the number of descendent genes with a level of confidence higher than 0.7 and the last column supplies additional biological information about the gene/ORF.

Confidence	Gene 1	Gene 2	Notes
1.0	YKL163W-PIR3	YKL164C-PIR1	Close locality on chromosome
0.985	PRY2	YKR012C	Close locality on chromosome
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 involved in Cytokinesis
0.92	YOR263C	YOR264W	Close locality on chromosome
0.91	YGR086	SIC1	Homolog to mammalian nuclear ran protein, both involved in nuclear function
0.9	FAR1	ASH1	Both part of a mating type switch, **expression uncorrelated**
0.89	CLN2	SVS1	Function of SVS1 unknown
0.88	YDR033W	NCE2	Homolog to transmembrame proteins suggest 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

Table 10.2: Source [12]. List of top Markov relations. The first column describes the level of confidence of the relation, the next two columns contain the names of the two genes of the relation and the last column supplies additional biological information referring the relation.

Bibliography

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

[2] T. Blumenthal. Gene clusters and polycistronic transcription in eukaryotes. *Bioessays*, pages 480–487, 1998.

[3] D. M. Chickering. A transformational characterization of equivalent Bayesian network structures. *Conference on Uncertainty in Artificial Intelligence (UAI '95)*, pages 87–98, 1995.

[4] D. M. Chickering. Learning bayesian networks is NP-complete. *Learning from Data: Artificial Intelligence and Statistics. Springer Verlag*, 5, 1996.

[5] G. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. *Proc. Fifthteenth Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pages 116–125, 1999.

[6] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[7] F. Cvrckova and K. Nasmyth. Yeast g1 cyclins cln1 and cln2 and a gap-like protein have a role in bud formation. *EMBO*, 12:5277–5286, 1993.

[8] M. A. Drebot, G. C. Johnston, J. D. Friesen, and R. A. Singer. An impaired rna polymerase ii activity in *saccharomyces cerevisiae* causes cell-cycle inhibition at start. *Mol Gen Genet*, 241:327–334, 1993.

[9] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, London, 1993.

[10] P.T. Spellman et al. Comprehensive identification of cell cycle-regulated genes of the yeast *sacccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[11] T. R. Hughes et al. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

[12] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

[13] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pages 139–147, 1998.

[14] N. Friedman, I. Nachman, and D. Pe'er. Bayesian network structure from massive datasets: The "sparse candidate" algorithm. *Proc. Fifthteenth Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pages 196–205, 1999.

[15] N. Friedman and Z. Yakhini. The sample complexity of learning bayesian networks. *Proc. Twelfth Conference on Uncertainty in Artificial Intelligence (UAI '96)*, pages 274–282, 1996.

[16] V. Guacci, D. Koshland, and A. Strunnikov. A direct link between sister chromatid cohesion and chromosome condensation revealed through the analysis of mcd1 in *s. cerevisiae*. *Cell*, 91:47–57, 1997.

[17] D. Heckerman. A tutorial on learning with bayesian networks. *Learning in Graphical Models*, 1998.

[18] D. Heckerman and D. Geiger. Learning bayesian networks: a unification for discrete and gaussian domains. *Proc. Eleventh Conference on Uncertainty in Artificial Intelligence (UAI 95)*, pages 274–284, 1995.

[19] D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[20] D. Heckerman, C. Meek, and G. Cooper. A bayesian approach to causal discovery. *Technical Report MSR-TR-97-05, Microsoft Research*, 1997.

[21] F. V. Jensen. An introduction to Bayesian networks. *University College London Press*, 1996.

[22] S. L. Lauritzan and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, 17:31–57, 1989.

[23] W. G. McGregor. DNA repair, DNA replication, and uv mutagenesis. *Investig Dermatol Symp Proc*, 4:1–5, 1999.

[24] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Francisco, Calif., 1988.

[25] J. Pearl and T. S. Verma. A theory of inferred causation. *Principles of Knowledge Representation and Reasoning: Proc. Second International Conference (KR '91)*, pages 441–452, 1991.

[26] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Proc. Ninth International Conference on Intelligent Systems for Molecular Biology (ISMB '01)*, 2001.

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

[28] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search.* Springer-Verlag, 1993.

[29] S. Tornaletti and P. C. Hanawalt. Effect of DNA lesions on transcription elongation. *Biochimie*, 81:139–146, 1999.

[30] `http://gened.emc.maricopa.edu/bio/bio181/BIOBK/BioBookgloss.html/`.

[31] `http://genomics.phrma.org/`.

[32] `http://mathworld.wolfram.com/MarkovChain.html/`.

[33] `http://www.cs.huji.ac.il/labs/compbio/expression/`.

[34] `http://www.cs.huji.ac.il/~pmai/class3/Class3.ppt/`.

[35] `http://yoda.cis.temple.edu:8080/UGAIWWW/lectures/bnets.html/`.

[36] G. Yona, N. Linial, and M. Linial. Protomap - automated classification of all protein sequences: a hierarchy of protein families, and local maps of the protein space. *Proteins: Structure, Function, and Genetics*, 37:360–378, 1998.