Using Bayesian Networks to Analyze Expression Data

Nir Friedman* Michal Linial† Iftach Nachman‡ Dana Pe'er§
Hebrew University
Jerusalem, 91904, ISRAEL

Abstract

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

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 graph-based model of joint multi-variate 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. cerevisiae* cell-cycle measurements of Spellman et al. [35] 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, due to regulation. 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 phase. By using DNA microarrays researchers are now able to measure the abundance of thousands of mRNA targets simultaneously [12, 27, 39]. 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 [24, 35].

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 them.

Most of the analysis tools currently used are based on *clustering* algorithms. These algorithms attempt to locate groups of genes that have similar expression patterns over a set of experiments [2, 4, 15, 29, 35]. Such analysis 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 [1, 6, 33, 38]. 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 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* [31]. 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.

*School of Computer Science and Engineering, nir@cs.huji.ac.il
†Institute of Life Sciences, michall@leonardo.ls.huji.ac.il
‡Center for Neural Computation & School of Computer Science and Engineering, iftach@cs.huji.ac.il
§School of Computer Science and Engineering, danab@cs.huji.ac.il

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 are well understood and have been used successfully in many applications. 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*. [22, 32, 36].

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 the gene-expression data of Spellman et al. [35], 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 Representing Distributions with 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 $\mathrm{Val}(X_i)$. 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 $\mathbf{x}, \mathbf{y}, \mathbf{z}$. 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.

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

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

where $\mathbf{Pa}(X_i)$ is the set of parents of X_i in G. Figure 1 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 (e.g., see [25, 31]), exploiting the explicit representation of structure in order to answer queries efficiently.

2.2 Equivalence Classes of Bayesian Networks

A Bayesian network structure G implies a set of independence assumptions in addition to (*). 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 (*).

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 cannot distinguish between equivalent graphs. Results of [7, 32] 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 arcs.

Theorem 2.1 *[32] Two graphs are equivalent if and only if their DAGs have the same underlying undirected graph and the same v-structures (i.e. converging directed edges into the same node, such as $a \rightarrow b \leftarrow c$).*

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 directed graph G, the PDAG representation of its equivalence class can be constructed efficiently [7].

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.) 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 score* (see [10, 21] for complete description):

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

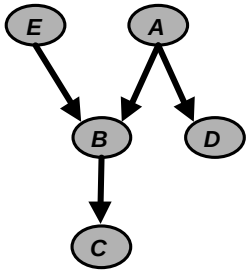

Figure 1: An example of a simple Bayesian network structure.

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 has the product form

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

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 [19]. 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. [21] present a family of priors, called *BDe priors*, that satisfy two important requirements: First, these priors are *structure equivalent*, i.e., 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

$$\mathrm{S}_{\mathrm{BDe}}(G : D) = \sum_i \mathrm{ScoreContribution}_{\mathrm{BDe}}(X_i, \mathbf{Pa}(X_i) : D),$$

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 (we use an *un-informative* prior in our experiments) and the data is given, learning amounts to finding the structure G that maximizes the score. This problem is known to be NP-hard [8], thus we resort to heuristic search. The decomposition of the score is crucial for this optimization problem. 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.

2.4 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. 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 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 casual 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. For example, 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 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 [22, 32, 36]. From observations alone, we cannot distinguish between causal networks that specify the same independence assumptions, i.e., belong to the same 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 arc $X \rightarrow Y$ 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. We stress that we can infer such causal relations without any experimental intervention (e.g. knockout and over-expressions) among our samples.

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 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 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 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 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 many variables, 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 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).[1] Our approach is 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. While at this point we handle only pairwise features, it is clear that this analysis is not restricted to them, and in the future we are planning on examining more complex features.

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 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 [31]. 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 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 global property. Recall that under the assumptions of Section 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.

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).

An effective, and relatively simple, approach for estimating confidence is the *bootstrap* method [14]. 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 effect 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 dataset.
  - Apply the learning procedure on D_i to induce a network structure $\hat{G}_i$.
- For each feature f of interest calculate

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

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

We refer the reader to [16] 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 2.3, 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 & 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 [18]. 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 algorithms 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. We assign each X_j some score of relevance to X_i, choosing variables with the highest score. A natural score that measures the dependence between two variables is their *mutual information* denoted $I(X;Y)$. The following is an example that arises with such a score: Consider the network in Figure 1. If $I(B;D) > I(B;E)$, for $k = 2$, E will be left out of C_B^n. Since A mediates the dependence between B and D, the network learned in this iteration will contain only A as B's parent. We can use this conditional independence to improve

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

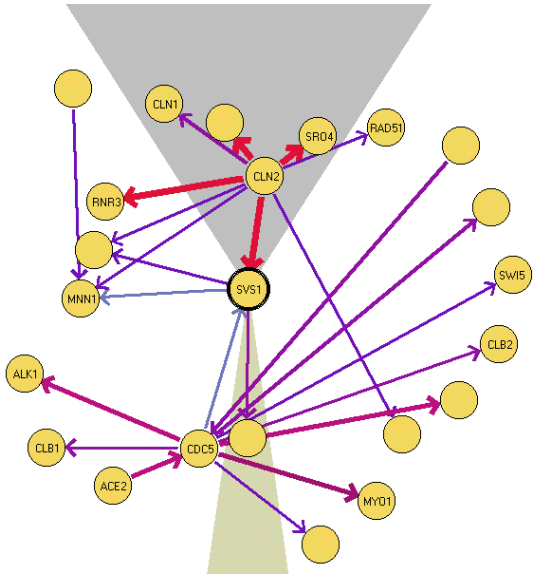

Figure 2: An example of the graphical display of Markov features. This graph shows a "local map" for the gene SVS1. The width (and color) of edges corresponds to the computed confidence level. An edge is directed if there is a sufficiently high confidence in the order between the pair genes connected by the edge. This local map shows that CLN2 seperates 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.

our candidate sets. A better score is the *conditional mutual information*, $I(X_i; X_j | \mathbf{Pa}_i^{G_n - 1}(X_i))$. The score we actually use is an estimator of the conditional mutual information in the underlying distribution, that takes into account also the number of parameters needed to learn X_i's conditional probability.

We refer the reader to [18] 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 whether the expression rate 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 [12]), 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. [23, 26, 30]). 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. [35], 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. [35] 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 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 additional variable denoting the cell cycle phase. This variable is forced to be a root in all the networks learned. Its presence allows to model dependency of expression levels on current cell cycle. [2]

We used the Sparse Candidate algorithm with a 200-fold bootstrap in the learning process. 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 conveyed in the measurements themselves. These results are available at our WWW site: http://www.cs.huji.ac.il/labs/compbio/expression. Figure 2 illustrates the graphical display of results of this analysis.

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 set 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. The runs on the random data sets do not learn almost anything with a confidence level above 0.8, which leads 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, we also 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 datasets. Figure 4 compares feature confidence in the analysis of the two datasets. As we can see, there is a strong correlation between confidence levels of the features between the two data sets.

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

[2] We note that we can learn temporal models using a Bayesian network that includes gene expression values in two (or more) consecutive time points [17]. This raises the number of variables in the model. We are currently perusing this issue.

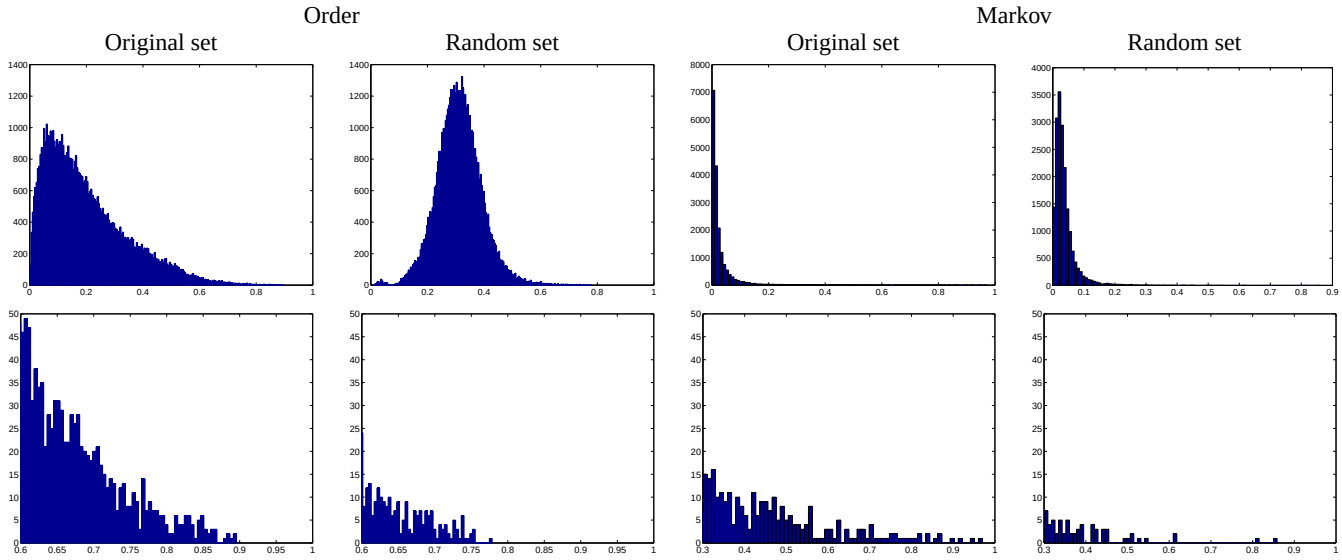

Figure 3: Histograms of confidence levels for the cell cycle data set, and the randomized data set. The histograms on the left are of order relations, and the ones on the right are of Markov relations. The histograms on the top row show the distribution of confidence levels in the interval $[0, 1]$. The histograms on the bottom row show the tails of these distributions for high-confidence features. These histograms are all based on the 250 genes data set.

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. Obviously, this linear correlation gets weaker for larger threshold differences. We also note that order relations are much more robust to changes in the threshold than the Markov relations.

A valid criticism of our discretization method is that it penalizes genes whose natural range of variation is small: since we use a fixed threshold, we would not detect changes in such genes. A possible way to avoid this problem is to *normalize* the expression of genes in the data. That is, we rescale the expression level of each gene, so that the relative expression all genes have the same mean and variance. We note that analysis methods that use *pearson correlation* to compare genes, such as [4, 15], are implicitly performing such a normalization. [3] When we discretize a normalized dataset, we are essentially rescaling the discretization factor differently for each gene, depending on its variance in the data. We tried this approach with several discretization levels, and got results comparable to our original discretization method. The 20 top Markov relations highlighted by this method were a bit different, but interesting and biologically sensible in their own right. The order relations were again more robust to the change of methods and discretization thresholds. A possible reason is that order relations depend on the network structure in a global manner, and thus can remain intact even after many local changes to the structure. The Markov relation, being a local one, is more easily disrupted. Since the graphs learned are extremely sparse, each discretization method "highlights" different signals in the data, which are reflected in the Markov relations learned.

In summary, although many of the results we report below (especially order relations) are stable across the different experiments discussed in the previous paragraph, it is clear that our analysis is sensitive to the discretization method. In all the different discretization methods we tried, our analysis found interesting relationships in the data. Thus, the challange is to find alternative methods that can recover all these relationships in one analysis. We are currently working on learning with (semi)parametric density models that would circumvent the need for discretization.

[3] An undesired effect of such a normalization is the amplification of measurement noise. If a gene has fixed expression levels across samples, we expect the variance in measured expression levels to be noise either in the experimental conditions or the measurements . When we normalize the expression levels of genes, we loose the distinction between such noise and true (i.e., significant) changes in expression levels. In our experiments, we can safely assume this effect will not be too grave, since we only focus on genes that display significant changes across experiments.

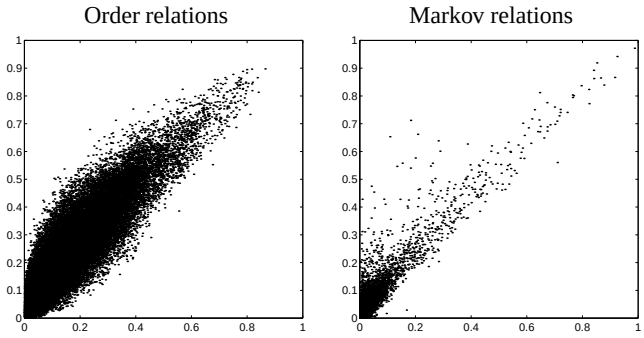

Figure 4: Comparison between significance levels with different number of genes in the analysis. Each relation is shown as a point, with the x-coordinate being its confidence in the the 250 genes data set and the y-coordinate the confidence in the 800 genes data set. The left figure shows order relation features, and the right figure shows Markov relation features; .

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.

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

Gene/ORF	Dominance Score	# of descendent genes $> .8$	$> .7$	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	
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

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. These dominant 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 dominating 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. For example, CLN1, CLN2 and CDC5, whose functional relation has been established [11, 13]. Other genes, like MCD1 and RFA2, were found to be essential [20]. 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 [28, 37].

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.

4.2.2 Markov Relations

Inspection of the top Markov relations reveals that most are functionaly related. A list of the top scoring relations can be found in table 2. Among these, all involving two known genes make sense biologically. When one of the ORFs is unknown careful searches using Psi-Blast [3], Pfam [34] and Protomap [40] 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 maintence protein). Several of the unknown pairs are physically adjacent on the chromosome, and thus presumably regulated by the same mechanism (see [5]), 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. Such analysis raises the number of biologically sensible pairs to 19/20.

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 [35] 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 [35] 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 and early cell cycle control, while there is no clear biological relationship between the others.

5 Discusion and Future Work

In this paper we presented a new approach for analyzing gene expression data that builds on the theory and algorithms for learning Bayesian networks. We described how to apply these techniques to gene expression data. The approach builds on two techniques that were motivated by the challanges posed by this domain: a novel search algorithm [18] and an approach for estimating statistical confidence [16]. We applied our methods to real expression data of Spellman et al. [35]. 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 [2, 4, 15, 29, 35], 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 [1, 6, 33, 38]. 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

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	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 uncorelated**
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

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* ([17]) 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 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 [9] 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 Gill Bejerano, Hadar Benyaminy, David Engelberg, Moises Goldszmidt, Daphne Koller, Matan Ninio, Itzik Pe'er, and Gavin Sherlock for comments on drafts of this paper and useful discussions relating to this work. We also thank Matan Ninio for help in running and analyzing the robustness experiments. This work was supported through the generosity of the Michael Sacher Trust.

References

[1] S. Akutsu, T. Kuhara, O. Maruyama, and S. Minyano. Indentification of gene regulatory networks by strategic gene disruptions and gene over-expressions. In *Proc. Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM-SIAM, 1998.

[2] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. 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, 1999.

[3] 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.

[4] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6:281–297, 1999.

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

[6] T. Chen, V. Filkov, and S. Skiena. Identifying gene regulatory networks from experimental data. In *Proc. 3'rd Annual International Conference on Computational Molecular Biology (RECOMB)*, 1999.

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

[8] D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics V*. Springer Verlag, 1996.

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

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

[11] F. Cvrckova and K. Nasmyth. Yeast G1 cyclins CLN1 and CLN2 and a GAP-like protein have a role in bud formation. *EMBO J*, 12:5277–5286, 1993.

[12] J. DeRisi., V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 282:699–705, 1997.

[13] 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.

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

[15] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Nat. Acad. Sci. USA*, 95:14863–14868, 1998.

[16] N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with Bayesian networks: A bootstrap approach. In *Proc. Fifthteenth Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pp. 206–215, 1999.

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

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

[19] N. Friedman and Z. Yakhini. On the sample complexity of learning Bayesian networks. In *Proc. Twelfth Conference on Uncertainty in Artificial Intelligence (UAI '96)*, pp. 274–282. 1996.

[20] 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(1):47–57, October 1997.

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

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

[23] R. Hoffman and V. Tresp. Discovering structure in continuous variables using Bayesian networks. In *Advances in Neural Information Processing Systems 8 (NIPS '96)*. MIT Press, 1996.

[24] V.R. Iyer, M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee, J.M. Trent, L.M. Staudt, J. Hudson, M.S. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P.O. Brown. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.

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

[26] D. Koller, U. Lerner, and D. Angelov. A general algorithm for approxiamte inference and its application to hybrid Bayes nets. In *Proc. Fifthteenth Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pp. 324–333, 1999.

[27] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Want, M. Kobayashi, H. Horton, and E. L. Brown. DNA expression monitoring by hybridization of high density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.

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

[29] G.S. Michaels, D.B. Carr, M. Askenazi, S. Fuhrman, X. Wen, and R. Somogyi. Cluster analysis and data visualization for large scale gene expression data. In *Pac. Symp. Biocomputing*, pp. 42–53. 1998.

[30] Kevin Murphy. Inference and learning in hybrid Bayesian networks. Technical Report CSD-98-990, U.C. Berkeley, 1998.

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

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

[33] R. Somogyi, S. Fuhrman, M. Askenazi, and A. Wuensche. The gene expression matrix: Towards the extraction of genetic network architectures. In *The Second World Congress of Nonlinear Analysts (WCNA)*, 1996.

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

[35] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *sacccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

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

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

[38] D. Weaver, C. Workman, and G. Stormo. Modeling regulatory networks with weight matrices. In *Pac. Symp. Biocomputing*, pp. 112–123, 1999.

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

[40] G. Yona, N. Linial, and Linial M. 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.