

Problems for Structure Learning: Aggregation and Computational Complexity

Frank Wimberly

Carnegie Mellon University (retired), USA

David Danks

Carnegie Mellon University and Institute for Human & Machine Cognition, USA

Clark Glymour

Carnegie Mellon University and Institute for Human & Machine Cognition, USA

Tianjiao Chu

University of Pittsburgh, USA

ABSTRACT

Machine learning methods to find graphical models of genetic regulatory networks from cDNA microarray data have become increasingly popular in recent years. We provide three reasons to question the reliability of such methods: (1) a major theoretical challenge to any method using conditional independence relations; (2) a simulation study using realistic data that confirms the importance of the theoretical challenge; and (3) an analysis of the computational complexity of algorithms that avoid this theoretical challenge. We have no proof that one cannot possibly learn the structure of a genetic regulatory network from microarray data alone, nor do we think that such a proof is likely. However, the combination of (i) fundamental challenges from theory, (ii) practical evidence that those challenges arise in realistic data, and (iii) the difficulty of avoiding those challenges leads us to conclude that it is unlikely that current microarray technology will ever be successfully applied to this structure learning problem.

INTRODUCTION

An important goal of cell biology is to understand the network of dependencies through which genes in a tissue type regulate the synthesis and concentrations of protein species. A mediating step in such synthesis is the production of messenger RNA (mRNA). Protein products of one gene may help to regulate the rate of transcription into mRNA of the DNA reading frame of certain other genes. These dependencies among gene activities and their mRNA proxies have long been represented by directed graphs. Early in the 1990s, machine learning algorithms were developed for learning directed graphs representing causal relations from appropriate data samples. At about the same time, developments in microarray techniques made possible the simultaneous measurement of messenger RNA (mRNA) counts for thousands of distinct genes. This juxtaposition naturally led to a flood of studies in the computer science and biological

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

literatures applying various search algorithms to gene expression data, with the aim of producing directed graphs that describe, for a tissue type, which genes regulate transcription rates of which other genes. Some of that work continues. We now know that the machine learning techniques are inappropriate and unsound in these applications, although they are potentially applicable to more recent measurements of RNA transcript concentrations in single cells. This chapter explains the statistical reasons why, as well as some of the relevant issues of computational complexity.

The short story is this: The goal of inference is the regulatory network within individual cells, but current microarray measurements are of mRNA counts extracted from large samples of cells. The machine learning algorithms exploit assumed symmetries between the network structure and a class of statistical properties of measurements. Assuming those symmetries hold for mRNA concentrations in individual cells and the regulatory network in the individual cellular level, and assuming all cells in the measured sample have the same regulatory network, it follows that the symmetry fails for measurements of concentrations aggregated from multiple cells. Experimental studies with real and simulated data confirm this failure.

THEORY: LEARNING FROM AGGREGATIONS

Microarrays are small chips a few square inches in size on which spots of DNA have been imbedded. A typical chip may contain thousands of spots, each spot composed of multiple copies of a small sequence of DNA. In the living cell nucleus, sections of DNA are copied (“transcribed”) into a dual complementary molecule, RNA, which is the scaffolding for the synthesis, outside the cell nucleus, of cellular proteins. RNA can be extracted from tissue, and tiny luminescent beads can be chemically attached to RNA molecules obtained from tissue cells (e.g., from breast cancer cells). Each RNA molecule contains a sequence of bases that binds to a specific DNA sequence. When a suspension consisting of many RNA molecules from a tissue sample is applied to a microarray, the RNA molecules bind to the complementary DNA sites. By measuring the luminosity of each DNA spot, the relative concentration of each kind of RNA in the tissue sample can be estimated. From these concentrations, one can infer relative activity of genes—how much RNA is produced by various parts of the cell DNA in the tissues sampled.

Two fundamentally different strategies have been proposed to determine networks of regulatory relationships from microarray measurements. One strategy (Yuh *et al.*, 1998; Ideker *et al.*, 2001; Davidson *et al.*, 2002) experimentally suppresses (or enhances) the expression of one or more genes, and measures the resulting increased or decreased expression of other genes. The method, while laborious, has proved fruitful in unraveling small pieces of the regulatory networks of several species. Its chief disadvantage is that each experiment provides information only about the effects of the manipulated gene or genes. A single knockout of gene *A* resulting in changed expression of genes *B* and *C*, for example, does not of itself provide information as to whether *A* regulates both *B* and *C* directly, or whether *A* regulates *B* which in turn regulates *C*, or whether *A* regulates *C* which in turn regulates *B*. And if manipulation of *A* yields no association with *B*, then *B* may still influence *A* (or not). This implies that at least $N-1$ experiments that intervene on a single gene would be required to identify the dependency structure of N genes, assuming those genes have no additional, unmeasured common causes, and this laborious procedure still cannot distinguish between direct and indirect regulation. Statistically, one uses only the estimation of the expression level of each gene considered in each experiment, and the uncertainties of those estimates. Experiments with multiple simultaneous interventions on gene expression complicate matters in ways we will discuss later.

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

A second strategy relies on the natural variation of expression levels of the same gene in different cells. The proposal is to measure—typically with microarrays—the expression levels in repeated samples from the same tissue source, or similar sources, and to infer the regulatory structure from the statistical dependencies and independencies among the measured expression levels (Akutsu *et al.*, 1998; D’haeseleer, 2000; D’haeseleer *et al.*, 2000; Friedman *et al.*, 2000; Hartemink, 2001; Hashimoto *et al.*, 2004; Liang *et al.*, 1998; Shmulevich *et al.*, 2002; Shrager *et al.*, 2002; Yoo *et al.*, 2002). The apparent advantage of the strategy is that it offers the possibility of determining multiple relationships without separate experimental interventions. If, for example, gene *A* regulates gene *C* only by regulating gene *B* which in turn regulates *C*, the expression level of *A* should be independent, or nearly independent, of the expression level of gene *C* conditional on the expression level of gene *B*. In principle, if adequate sample sizes were available, the methods could also be used as a supplement to gain additional information from experiments in which the expression of particular genes are experimentally suppressed or enhanced (but see the Computational Complexity section below). The requisite statistical procedures for this strategy are more elaborate, and require direct or indirect (e.g. implicit in the posterior probabilities) estimates of conditional independence relationships among expression levels.

There are many statistical obstacles to the second strategy including: the joint influence of unmeasured factors (e.g. unmeasured gene expressions or extra-cellular factors), a variety of sources of measurement error, an unknown family of probability distributions governing the errors, and functional dependencies for the expression of any gene that may be Boolean for some regulating genes and continuous for other regulators. Some of these difficulties—in particular the presence of unrecorded common causes—can, in principle, be overcome (Spirtes *et al.*, 2001). We describe in this section a more elementary statistical difficulty with the second strategy that calls its value into question and raises a set of important research problems.

Directed Acyclic Graphs and Markov Factorization

Qualitative regulatory relationships among genes are often represented by directed graphs. Each vertex is a random variable whose values represent levels of expression of a particular gene. Each directed edge from a variable *X* to a variable *Y* in such a graph indicates that *X* produces a protein that regulates *Y*. In principle, the graph may be cyclic or acyclic, and may even have self-loops (a directed edge from a variable to itself). In the simplest case, one assumes an acyclic graph with noises and random measurement errors for each measurement of each gene that are independent of those for any other gene.

We consider this simplest case: the true, but unknown regulatory structure can be represented by a directed acyclic graph, with independent errors. Consider, for example, four genes *X*, *Y*, *Z*, *W* whose regulatory connections can be represented by Figure 1.

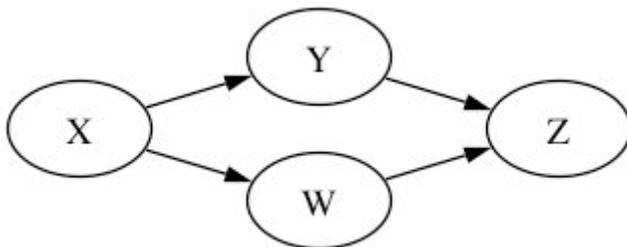


Figure 1: Example directed acyclic graph

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

Suppose the measured values of X, Y, Z, W satisfy the following three equations:

$$Z = f(Y, W) + \varepsilon_Z$$

$$Y = g(X) + \varepsilon_Y$$

$$W = h(X) + \varepsilon_W$$

f, g, h are any functions and $\varepsilon_Z, \varepsilon_Y, \varepsilon_W$ are independently distributed noises. It follows that the joint probability density of X, Y, Z, W admits a Markov factorization: $d(X, Y, Z, W) = d(Z | Y, W) d(Y | X) d(W | X) d(X)$. The Markov factorization implies that Y and W are independent conditional on X , and that X and Z are independent conditional on $\{Y, W\}$; it is in fact equivalent to specifying that these two relationships hold. More generally, assuming each random variable has an independent noise source but is otherwise a deterministic function of its parents in the graph, the system described by any directed acyclic graph has a density that admits a Markov factorization that can be written as the product, over all variables, of the density of each variable conditional on its graphical parents. Graphs with the same Markov factorization imply the same independencies and conditional independencies, and so form an equivalence class. The Markov equivalence class for Figure 1 consists of that graph and the graphs obtained by reorienting exactly one of: $X \rightarrow Y$ or $X \rightarrow W$. The Markov equivalence class represents the most information that could be obtained from second moments of the joint distribution of the variables. Non-Normal distributions have higher moments that are not uniquely determined by the second moment, and for linear systems it has been shown that higher moments can resolve structure more finely than the Markov equivalence class. We focus here on the dominant type of search algorithm for gene regulation networks: namely, those that assume either linearity or rely exclusively on second moments, even though some independently established expression dependencies are known to be non-linear. For time series data, regulatory relationships can still be represented by a directed acyclic graph and probabilities admitting a Markov factorization, but with vertices appropriately labeled by gene and time.

The Challenge of Aggregation

In structure learning, the aim is to discover the regulatory structure in individual cells, but measurements are typically of relative concentrations of mRNA transcripts obtained from thousands, or even millions, of cells. Such measurements are not of variables such as X in Figure 1, but are instead, ideally, of the sum of the X values over many cells. We will denote such measured sums over n cells by ΣX_i .

In general, the conditional dependencies/independencies among the gene expression levels of a single cell are not the same as those among the sums of gene expression levels over a number of cells. This statistical fact poses a serious difficulty for the second strategy for regulatory structure inference, which relies on the statistical dependencies among the gene expression levels. For example, if the variables in Figure 1 are binary, and each measurement is of the aggregate of transcript concentrations from two or more cells, ΣX_i and ΣZ_i are not independent conditional on $\{\Sigma Y_i, \Sigma W_i\}$, and the associations obtained from repeated samples will not therefore satisfy the Markov factorization (Danks & Glymour, 2002).

There are some special cases where the conditional independencies are invariant under aggregation. For example, if binary regulatory relations among genes X, Y , and Z are described by a singly connected graph such as $X \rightarrow Y \rightarrow Z, X \leftarrow Y \leftarrow Z$, or $X \leftarrow Y \rightarrow Z$, then the implied conditional independence of X and Z given Y holds as well for sums of independent measurements of X, Y , and Z respectively (Danks & Glymour, 2002).

Linear, Normal distributions have special virtues for invariance. Whatever the directed acyclic graph of cellular regulation may be, if each variable is a linear function of its parents and an independent Gaussian noise, then the Markov factorization holds for the summed variables. In that case, conditional independence is equivalent to vanishing partial correlation, and the partial correlation of the two variables (although not the sampling distribution), each respectively composed of the sum of n like variables, will be the same as the partial correlation of the unsummed variables.

Two less restrictive sufficient conditions for conditional independence of variables to be the same as the conditional independence of their sums, are given in two theorems proved in Chu, *et al.* (2003):

Theorem 1 (Local Markov theorem): Given an acyclic graph G representing the causal relations among a set \mathbf{V} of random variables, let $Y, X_1, \dots, X_k \in \mathbf{V}$, and $\mathbf{X} = \{X_1, \dots, X_k\}$ be the parents of Y in G . If $Y = \mathbf{c}^T \mathbf{X} + \varepsilon$, where $\mathbf{c}^T = (c_1, \dots, c_k)$, and ε is a noise term independent of all non-descendants of Y , then Y is independent of all its non-parents, non-descendants conditional on its parents \mathbf{X} , and this relation holds under aggregation.

Theorem 2 (Markov wall theorem): Given an acyclic graph G representing the causal relations among a set \mathbf{V} of random variables. Let $\mathbf{X} = \{X_1, \dots, X_h\}$, $\mathbf{Y} = \{Y_1, \dots, Y_k\}$, $\mathbf{W} = \{W_1, \dots, W_m\}$, and $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{W} = \mathbf{V}$. Suppose that the following three conditions hold:

- 1) The joint distribution of $X_1, \dots, X_h, Y_1, \dots, Y_k$ is multivariate normal with non-singular covariance matrix;
- 2) For $i = 1, \dots, k$, Y_i is neither a parent nor a child of any variable $W_j \in \mathbf{W}$ (i.e., there is no direct edge between a variable in \mathbf{Y} and a variable in \mathbf{W}); and
- 3) For $i = 1, \dots, h$, X_i is not a child of any variable $W_j \in \mathbf{W}$ (i.e., any edge between a variable in \mathbf{X} and a variable in \mathbf{W} must be from the \mathbf{X} -variable to the \mathbf{W} -variable).

Then \mathbf{Y} is independent of \mathbf{W} conditional on \mathbf{X} , and this relation holds under aggregation.

Although there are established regulatory mechanisms in which some regulators of a gene act linearly in the presence of a suitable combination of other regulators of the same gene (Yuh *et al.*, 1998), there does not appear to be any known regulatory system that is simply linear.

One of the best-established regulatory functional relations seems to be the expression of the Endo16 gene of the sea urchin (Yuh *et al.*, 1998). The expression level of the gene is controlled by a Boolean regulatory switch between two functions, each of which is a product of a Boolean function of regulator inputs multiplied by a linear function of other regulator inputs. Even much simplified versions of such transmission functions do not preserve conditional independence over sums of variables.

Consider an example: suppose in each of n cells genes X, Y, Z , and W have the regulatory structure $X \rightarrow Y \rightarrow Z \leftarrow W$ with $Y = X^2$; $Z = YW$; and W a binary variable with $P(W = 1) = p$. Assume X takes values in $\{0, 1, 2, 3, 4\}$ with uniform probability. Let $\Sigma X_i, \Sigma Y_i, \Sigma Z_i$, and ΣW_i denote the sums of values of X, Y, Z and W respectively over $n = 4$ cells. Z is independent of X given Y in each cell. However, we will show that ΣZ_i is not independent of ΣX_i given ΣY_i .

For each cell i , $Z_i = Y_i$ if the value of W_i is 1, and zero otherwise. Hence the probability that $Z_i = y_i$ given that $Y_i = y_i$ is $p + (1-p)/5$. Let $\Sigma Y_i = \Sigma (X_i)^2 = 16$. There are just five possible vector values for $\mathbf{X} = \langle X_1, X_2, X_3, X_4 \rangle$ consistent with $\Sigma (X_i)^2 = 16$: $\langle 4, 0, 0, 0 \rangle$; $\langle 0, 4, 0, 0 \rangle$; $\langle 0, 0, 4, 0 \rangle$; $\langle 0, 0, 0, 4 \rangle$ and $\langle 2, 2, 2, 2 \rangle$. The first four vectors in the list have $\Sigma X_i = 4$ and the last has $\Sigma X_i = 8$. We will now show that $P(\Sigma Z_i = 16 \mid \Sigma Y_i = 16 \ \& \ \Sigma X_i = 4)$ is not in general equal to $P(\Sigma Z_i = 16 \mid \Sigma Y_i = 16 \ \& \ \Sigma X_i = 8)$. For example, if $\mathbf{X} = \langle 4, 0, 0, 0 \rangle$, then $\Sigma Z_i = 16$ if and only if $W_1 = 1$, where $P(W_1 = 1) = p$. Similarly for the vectors $\langle 0, 4, 0, 0 \rangle$, $\langle 0, 0, 4, 0 \rangle$ and $\langle 0, 0, 0, 4 \rangle$. Given that $\Sigma X_i = 4$ and

$\sum Y_i = \sum (X_i)^2 = 16$, the set of the first four vectors has probability 1, and each individual vector of the first four has probability 0.25. Therefore $P(\sum Z_i = 16 \mid \sum Y_i = 16 \ \& \ \sum X_i = 4) = p$. On the other hand, the probability that $\mathbf{X} = \langle 2, 2, 2, 2 \rangle$ is 1 given that $\sum X_i = 8$ and $\sum Y_i = \sum (X_i)^2 = 16$. Therefore $P(\sum Z_i = 16 \mid \sum Y_i = 16 \ \& \ \sum X_i = 8)$ is just the probability that $W_i = 1$ for $i = 1, 2, 3, 4$, which is p^4 .

Much about the preceding example—e.g., that $n = 4$, that X is uniformly distributed, that X has 5 distinct values, that $Y = X^2$ —is obviously inessential; $Y = X^2$ was used only because it is the simplest non-linear, non-Boolean function proposed for a regulator (Schilstra, 2002). Similar arguments would apply to a variety of non-linear dependencies of Y on X .

The considerations we have advanced in this section argue that, other than by chance, inference to genetic regulatory networks from associations among measured expression levels is possible only if conditional independence relations in the individual cells are (approximately) preserved in sums of those i.i.d. units. Although the particular example we gave was not biologically relevant, there are biologically relevant cases in which those conditional independence relations are not preserved. Chu (2004) has provided general sufficient conditions for conditional independence relations not to be invariant.

There are conditions under which the conditional independence relations among the summed expression levels of the genes from large number of cells will eventually be determined by the covariance matrix of the expression levels of the genes within a single cell. Recall that unlike conditional independence relations, the covariance matrix, with appropriate normalization, is invariant under aggregation. Those conditions are given in Theorem 3.

Theorem 3: Let $\{(X_n, Y_n, \mathbf{Z}_n)\}$ be a sequence of i.i.d. $k+2$ dimensional random vectors with mean $\mathbf{0}$ and nonsingular covariance matrix Σ . Suppose (X_n, Y_n, \mathbf{Z}_n) and \mathbf{Z}_n both have bounded densities (with respect to the Lebesgue measure). Let $X_n^* = \sum X / \sqrt{n}$, $Y_n^* = \sum Y / \sqrt{n}$, and $\mathbf{Z}_n^* = \sum \mathbf{Z} / \sqrt{n}$, and (U, V, \mathbf{W}) be a multivariate normal random vector with mean $\mathbf{0}$ and covariance matrix Σ . Then the total variation distance between (i) the conditional distribution of (X_n^*, Y_n^*) given \mathbf{Z}_n^* , and (ii) the product of the conditional distributions of X_n^* given \mathbf{Z}_n^* and Y_n^* given \mathbf{Z}_n^* converges to: the total variation distance between (i) the conditional distribution of (U, V) given \mathbf{W} , and (ii) the product of the conditional distributions of U given \mathbf{W} and V given \mathbf{W} almost surely with respect to the measure induced by \mathbf{W} .

The implication of Theorem 3 is that, assuming we can model the gene expression levels as continuous random variables satisfying some regularity conditions, the conditional independence relations among the summed expression levels of a large number of cells are determined by the covariance matrix of the summed expression levels, regardless of the conditional independence relations among the gene expression levels in a single cell. For example, if in a single cell, gene X and gene Y are independent given gene Z , but the partial correlation between X and Y given Z is non-zero—this is usually the case when (X, Y, Z) do not follow a multivariate normal distribution—then given n such cells, $\sum X$ and $\sum Y$ are dependent given $\sum Z$. (Note that the correlation matrix of the gene expression levels is preserved under aggregation.)

While the conditions for Theorem 3 seem to be quite general, they do not cover the class of discrete distributions. After all, the expression level of any gene in a cell—the number of mRNA transcripts for that gene at a moment—is an integer-valued random variable. Continuous distributions can approximate a discrete distribution arbitrarily well, though only in terms of the distribution function. Theorem 3 can, however, be extended to an important class of discrete distributions—the regular lattice distributions—which covers the possible distributions of the numbers of mRNA transcripts of any set of genes in a cell.

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

A lattice distribution for a random vector \mathbf{X} is a discrete distribution that only assigns non-zero probabilities to points $\mathbf{x} = (x_1, \dots, x_k)$ such that $x_i = mh_i + b_i$, where m is an integer, h_i a positive real value, and b_i a constant. If h_i is the largest positive real number such that X_i can only take values of the form $mh_i + b_i$, h_i is called the span of X_i . A regular lattice distribution is defined as: Suppose a random vector $\mathbf{X} = (X_1, \dots, X_k)$ has a lattice distribution, and h_i is the span of X_i . \mathbf{X} has a *regular* lattice distribution if, for each i , there are at least two vectors $\mathbf{x}^i = (x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_k)$ and $\mathbf{y}^i = (x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_k)$, such that $|y_i - x_i| = h_i$, $P(\mathbf{X} = \mathbf{x}^i) > 0$, and $P(\mathbf{X} = \mathbf{y}^i) > 0$.

Chu (2004) extended Theorem 3 to cases in which \mathbf{Z}_n has a regular lattice distribution, not a bounded density. These results provide a sufficient condition for the conditional independence relation between genes X and Y given genes \mathbf{Z} in a single cell *not* to be invariant under aggregation: namely, when the partial correlation of X and Y given \mathbf{Z} is non-zero. Notice, however, that the partial correlation is invariant under aggregation.

These theoretical results appear to conflict with many reports of successful machine learning searches for regulatory structure. In many cases, however, the successes are with simulated data in which the simulated values for individual cell representatives are not summed in forming the simulated measured values, and are therefore unfaithful to the actual measurement processes. In several other cases, results with real data are not independently confirmed, but merely judged plausible. Rarely, results are obtained that agree with independent biological knowledge; in these cases the actual regulatory structure among the genes considered may approximately satisfy invariance of conditional independence for summed variables, or the procedures may simply have been lucky. Feasible, economical techniques for measuring concentrations of transcripts in single cells could make machine learning techniques based on associations of expressions valuable in identifying regulatory structure. Techniques for the measurement of concentrations of mRNA species have recently become available (Levsky *et al.*, 2002; Elowitz *et al.*, 2002; Ginsberg *et al.*, 2004; Rosenfeld *et al.*, 2005), and seem the more appropriate venue for the application of machine learning methods

Experimental techniques that take advantage of immunoprecipitation, tagging of binding sites and regulatory proteins, binding site sequence homologies, and evolutionary preservation of regulatory mechanisms, are proving more fruitful. We may hope that machine learning techniques that are biased by such extra information may prove useful, and two recent examples in the literature suggest that this hope may bear fruit.

Pe'er *et al.* (2006) focus on learning the structure of what they term *regulation graphs*: those in which (i) a small subset of vertices are regulators; (ii) a non-regulator is not the parent of any other vertex; and (iii) the number of parents of any vertex is bounded by some small number. These graphical structures correspond to those in which there are a few regulators that control the activity of all other genes, and perhaps influence each other. The restriction to this relatively small set of possible graphs results in provable performance guarantees for their MinReg algorithm, but it is computationally intractable unless the set of possible regulators is small. Microarray data alone do not suffice to determine a small set of possible regulators, and so Pe'er *et al.* (2006) use additional sources of information—functional annotations from other experiments, and sequence homologies—to restrict the possible regulator set. When these additional pieces of information are used for real-world data, the MinReg algorithm recovers more information than other algorithms; insufficient information is provided to accurately judge its absolute performance. The simulation tests of the MinReg algorithm cannot be evaluated since they do not use aggregated data (see next section).

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

Hartemink (2006) similarly uses auxiliary data both to constrain the set of possible graphs and to provide a prior bias. Hartemink's algorithm is a standard Bayesian learning algorithm in which prior knowledge is incorporated through a bias in the prior probability over possible graphs. He focuses on two different settings: one is a (relatively) static system that can be modeled using an acyclic graph; the other is a dynamic system that is modeled using a dynamic Bayesian network (essentially, a graph in which the variables are time-indexed). Auxiliary information is used to restrict the possible graphs through both (i) variable selection, as the two systems have only 32 and 25 variables picked out by biological function; and (ii) graphical restrictions on the dynamic network, as the current time step is assumed to be directly influenced only by the previous time step. Given these sets of possible graphs, transcription factor binding location data are used to provide a significant bias on the prior probabilities over the graphs. In the actual applications to real-world data (judged by comparison to a gold-standard network), the location data play a major role: the algorithm performs quite poorly when expression data alone are used. In fact, close consideration of the real-world results suggests that the location data is doing almost all of the work. The algorithm's performance using both types of data is only marginally superior to using location data alone.

EXPERIMENTAL RESULTS

As elaborated in the previous section, microarray measurements are from aggregates of thousands of cells, and conditional independence relations that hold for biologically realistic probability distributions in individual units are typically not the same as those that hold in the probability distribution for cell aggregates. There are at least seven other challenges facing algorithms for automated learning of regulatory network structure: (1) the number of measurements of each gene is typically much smaller than the number of genes under study, and the number of genes—or genes at time points in time series representations—effectively defines the number of variables; (2) microarray measurements have a small signal to noise ratio; (3) many algorithms are based on acyclic graph representations which cannot faithfully represent both the probability distributions for equilibrium distributions of feedback systems and the mechanisms that lead to an equilibrium, and the only provably correct algorithm for learning arbitrary cyclic graphs from equilibrium data that is currently available (Richardson, 1996) has never been tested on gene expression data; (4) statistical associations among measured expression levels for different genes may depend on variations in unrecorded “common cause” regulator genes, or on extra-genetic factors not in the database; (5) summing variable values over many cell units reduces their variance, resulting in low correlations due to regulatory interaction, which implies the need for either very large samples or very large expression differences to reliably distinguish zero from non-zero correlations; (6) discretization of continuous variables can alter the original conditional independence relations among variables; and (7) when there are unrecorded sources of covariation, linear regression techniques overfit in the linear case, positing false connections even without correlated errors (Spirtes *et al.*, 2001), and feedback can produce statistical dependencies among measured variables similar to the effects of omitted common causes.

There are reports of successes at network inference with machine learning methods applied to both real-world and simulated expression data, but to our knowledge, no published simulation studies generate their data from experimentally established networks and treat measured values as aggregates of many individual cell values. In this section, we use realistic, aggregated, simulated data to examine the performance of nine algorithms: Reveal (Liang *et al.*, 1998), Bool2 (Akutsu *et al.*, 2000), MRBN (Friedman *et al.*, 1999), PC (Spirtes *et al.*, 2001), CCD (Richardson, 1996),

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

and algorithms described in Spirtes & Meek (1995), Arkin *et al.* (1997), D'Haeseleer *et al.* (1999), Weaver *et al.* (1999), and van Someren *et al.* (2000). The version of MRBN we use was implemented by Aaron Darling (see <http://mrbn.dyndns.org/>), as other downloadable versions did not run, reimplementations were not possible from published accounts, and the authors did not respond to requests for clarification. PC and CCD were obtained from <http://www.phil.cmu.edu/projects/tetrad>. The Meek/Spirtes algorithm was provided by Peter Spirtes from an old implementation not currently publicly available. We implemented the Reveal and Bool2 algorithms from published descriptions. The remaining algorithms were obtained from <http://genlab.tudelft.nl/info>.

These algorithms include procedures that discretize variables to binary or ternary values (Reveal, Bool2, MBRN), procedures that treat variables as continuous, procedures that use optimization routines (Bool2), regression procedures of various kinds (Weaver, van Someren, Arkin, D'Haeseleer), constraint based searches (PC, CCD), Bayesian scoring searches (MBRN) and hybrid constraint/Bayesian searches (Spirtes/Meek). Clearly these are not all of the algorithms that have been or could be proposed for studying gene regulation. For example, we have not applied the FCI algorithm (Spirtes *et al.*, 2001), nor have we included simulated annealing algorithms (Hartemink, 2001) or heuristic scoring procedures for Bayes nets with time indexed variables. We attempted to include a recent algorithm proposed and applied by Pe'er, *et al.* (2002), but the authors declined to provide their implementation.

This study used four datasets—three simulated, and one experimental:

1. Data generated in ten steps from a time series network modeling regulation in a fragment of the sea urchin genome (see Figure 2);
2. Data similar to (1) but projected to binary values;
3. Data similar to (1) but projected to three values;
4. Data from microarray measurements of variations of expression levels over the cell cycle in yeast (Spellman *et al.*, 1998) compared with a recent experimental determination of a substantial fraction of the regulatory network in the same species (Lee *et al.*, 2002).

The first three datasets are based on the multi-year effort by Davidson and his collaborators (Davidson *et al.*, 2002) to elucidate the genetic network of the sea urchin embryo, resulting in experimental data for a network of some forty genes. We developed a Java implementation of the “maternal and early interactions” portion of the sea urchin network, at least as it was understood at the time, using realistic transfer functions relating gene inputs to their outputs (see Figure 2; note that there are six genes: Wnt8, Krl, SoxB1, Krox, Otx, and Eve). This network has several feedback loops, including three genes that directly auto-regulate. We note in passing that the “truth” of the simulated network is irrelevant to the point at hand; all that matters is that this network involves realistic connectivity and realistic transfer functions. To simulate measurement noise, we multiplied the output value for each gene by the value of a random Gaussian variable with mean 1 and variance 0.01. We did not include additive error.

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

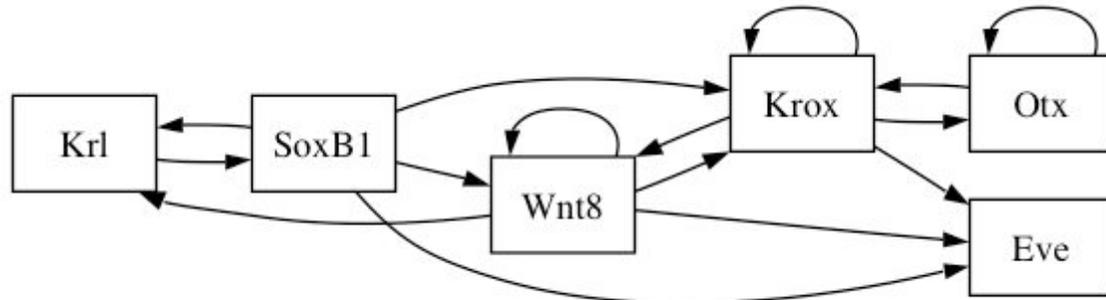


Figure 2: The “maternal and early interactions” portion of the regulatory network of the sea urchin embryo (Davidson *et al.*, 2002)

Our Java implementation realized a detailed reconstruction of the transfer functions and other features of the network as implemented in a NetBuilder model of the maternal and early interactions portion of the organism. NetBuilder is well documented (Schilstra, 2002), as is its model of this organism. It allows a user to “build” a gene network and to specify complex, non-linear and Boolean transfer functions. We built a NetBuilder version of the network under study and carried out a comparison of our code’s calculations with NetBuilder’s output over a number of steps; the results agreed closely. Christophe Battail has published a web page comparing NetBuilder’s simulation of the Endo16 gene with experimental results (<http://strc.herts.ac.uk/bio/maria/NetBuilder/Examples/Endo16/Endo16sim02.htm>).

To create a *non-aggregated dataset*, we recorded the simulated expression level for each of the six genes (Wnt8, Krl, etc.) at each of up to 10 time steps; we call the values recorded for one such run a *non-aggregated sample*. The data matrix thus has simulations in rows, and each gene-time step as a column (variable). In the description of results below, S denotes the number of non-aggregated samples in a particular non-aggregated dataset. To construct an *aggregated sample*, we compute the mean for each column of a non-aggregated dataset; each aggregated sample thus corresponds to a full, non-aggregated dataset. An *aggregated dataset* is a collection of such aggregated samples. In the results below, R denotes the number of non-aggregated datasets used to construct each aggregated sample in an aggregated dataset. In all of our experiments the sample sizes are comparatively small—reflecting the reality of microarray studies—and in most cases, the distributions are non-Gaussian, and the dependencies are non-linear. We approximated mean-zero normality by taking logs of all values in the data matrices (for both non-aggregated and aggregated datasets) and then subtracting the median of each column from all the values in that column.

By projecting based on the median value of each variable, we binarized the same data for tests of the Reveal and Bool2 algorithms. The MRBN algorithm implementation automatically projects real values to one of three values. The PC and CCD algorithms require multiple samples, each consisting of an entire time series. The binary algorithms require as input a dataset consisting of binary values for each of a set of genes at each of a number of time steps, and so the same time series can be used for comparisons. All datasets are publicly available at <http://www.phil.cmu.edu/projects/genegroup>.

The fourth dataset comes from four experiments (Spellman *et al.*, 1998) in which mRNA expression levels were measured in the course of the cell cycle with cells synchronized in different ways (see http://staffa.wi.mit.edu/cgi-bin/young_public/navframe.cgi?s=17&f=downloaddata). Friedman *et al.* (2000) applied the

MRBN algorithm to this data to obtain conjectured regulatory relations among the genes. Comparison experimental data are from Lee *et al.* (2002), who applied immunoprecipitation techniques to experimentally estimate genes directly regulated by each of more than 100 known yeast regulators. The different Spellman experiments used samples in different metabolic conditions, so it is not sensible to use them as repeated samples of the same time series. We therefore simply concatenated the data so they appeared to be from one experiment; this introduces 3 false breaks in time series.

PC, CCD and Spirtes/Meek take a significance level as input; we give results for .05, but results for other significance levels up to 0.3 are similar, with lower significance levels slightly better in most experiments. Results for the van Someren and D’Haeseleer algorithms were essentially identical and we show only the latter.

To assess the performance of the algorithms, we ignored edge direction and focused on only the (simpler) problem of determining adjacency relations. An adjacency is judged present between two genes in an algorithm output if and only if it is present between those genes for any two times. Many other counting procedures are possible within each experiment (e.g., majority rule; restriction to sequential time steps) that would reduce false positives and increase false negatives. There are twenty-one possible pairs of adjacent variables (since a gene can auto-regulate), and twelve of those pairs are actually adjacent in the true graph for the maternal and early interactions portion of the sea urchin embryo network. Random assignment of edges for pairs of genes would result in 10.5 expected errors and an error rate of 0.5. Simply saying “yes” to each possible adjacency would result in an error rate of 0.43. Tables 1-5 report mean performance (and variance in parentheses) over 10 replications of the simulation settings; note that the variance for false positives determines the variance of correct negatives (and similarly for false negatives and correct positives).

	False Pos	Correct Pos	False Neg	Correct Neg	Total Errors	Error rate
PC05	5.2 (1.3)	9.2	2.8 (0.6)	3.8	8.0 (2.2)	0.38
CCD05	4.4 (2.7)	7.6	4.4 (2.3)	4.6	8.8 (5.5)	0.42
Meek	4.2 (2.2)	10.4	1.6 (0.5)	4.8	5.8 (4.2)	0.28
Reveal	6.0 (0.1)	8.2	3.8 (0.2)	3.0	9.8 (0.4)	0.47
Bool2	6.1 (0.6)	7.9	4.1 (0.1)	2.9	10.2 (0.3)	0.49
MRBN	3.2 (1.0)	3.2	8.8 (2.6)	5.8	12 (3.7)	0.57
Arkin	1.4 (0.02)	2.7	9.3 (0.02)	7.6	10.7 (0.01)	0.51
Weaver	8.9 (0.01)	11.9	0.1 (0.0)	0.1	9.0 (0.01)	0.43
D’Haeseleer	5.4 (0.09)	7.3	4.7 (0.14)	3.6	10.1 (0.11)	0.48

Table 1. Non-aggregated datasets; $S = 20$ (i.e., 20 non-aggregated samples in each dataset)

	False Pos	Correct Pos	False Neg	Correct Neg	Total Errors	Error Rate
PC05	5.4 (1.8)	8.9	3.1 (2.1)	3.6	8.5 (2.5)	0.40
CCD05	4.9 (0.54)	8.5	3.5 (2.5)	4.1	8.4 (3.2)	0.40
Meek	5.7 (0.9)	9.1	2.9 (0.8)	3.3	8.6 (1.2)	0.41
Reveal	6.1 (0.0)	8.1	3.9 (0.07)	2.9	10 (0.17)	0.48
Bool2	6.3 (0.06)	7.8	4.2 (0.0)	2.7	10.5 (0.1)	0.50
MRBN	1.9 (1.0)	4.0	8.0 (0.4)	7.1	9.9 (2.1)	0.47

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

Arkin	1.4 (0.1)	2.7	9.3 (0.05)	7.6	10.7 (0.26)	0.51
Weaver	9.0 (0.0)	11.9	0.1 (0.0)	0.0	9.1 (0.0)	0.43
D'Haeseleer	5.3 (0.07)	7.0	5.0 (0.14)	3.7	10.3 (.34)	0.49

Table 2. Aggregated samples; $S = 20$ and $R = 30$ (i.e., 30 aggregated samples)

	False Pos	Correct Pos	False Neg	Correct Neg	Total Errors	Error Rate
PC05	5.7 (1.8)	9.3	2.7 (1.8)	3.3	8.4 (3.6)	0.40
CCD05	5.4 (0.7)	8.8	3.2 (2.2)	3.6	8.6 (4.0)	0.41
Meek	5.6 (0.9)	9.1	2.9 (1.9)	3.4	8.5 (4.5)	0.40
Reveal	6.1 (0.0)	8.2	3.8 (0.0)	2.9	9.9 (0.0)	0.47
Bool2	6.2 (0.0)	7.8	4.2 (0.0)	2.8	10.4 (0.0)	0.50
MRBN	2.2 (1.3)	3.7	8.3 (1.1)	6.8	10.5 (1.4)	0.50
Arkin	1.5 (0.04)	2.7	9.3 (0.09)	7.5	10.8 (0.2)	0.51
Weaver	9.0 (0.0)	12.0	0.0 (0.0)	0.0	9.0 (0.0)	0.43
D'Haeseleer	5.5 (0.18)	7.3	4.7 (.17)	3.5	10.2 (0.16)	0.49

Table 3. Aggregated samples; $S = 20$ and $R = 100$

	False Pos	Correct Pos	False Neg	Correct Neg	Total errors	Error rate
PC05	1.2 (1.1)	6.9	5.1 (1.4)	7.8	6.3 (2.7)	0.30
CCD05	1.4 (0.7)	7.3	4.7 (0.7)	7.6	6.1 (2.3)	0.29
Meek	1.8 (1.5)	7.5	4.5 (0.95)	7.2	6.3 (3.1)	0.30
Reveal	5.2 (0.0)	6.8	5.2 (0.0)	3.8	10.4 (0.0)	0.50
Bool2	5.3 (0.0)	7.6	4.4 (0.0)	2.7	9.7 (0.0)	0.46
MRBN	1.9 (0.5)	3.4	8.6 (1.4)	7.1	10.5 (1.2)	0.50
Arkin	1.2 (0.0)	3.2	8.8 (0.03)	7.8	10.0 (0.07)	0.48
Weaver	4.5 (0.17)	3.8	8.2 (0.1)	4.5	12.7 (0.02)	0.60
D'Haeseleer	8.0 (0.03)	11.3	0.7 (0.02)	1.0	8.7 (0.03)	0.41

Table 4. Non-aggregated samples; $S = 100$ and linear transfer functions were used

	False Pos	Correct Pos	False Neg	Correct Neg	Total errors	Error rate
PC05	1.2 (0.84)	7.4	4.6 (0.84)	7.8	5.8 (3.7)	0.28
CCD05	1.7 (0.9)	7.5	4.5 (1.6)	7.3	6.2 (3.0)	0.30
Meek	2.0 (0.9)	7.5	4.5 (0.7)	7.0	6.5 (1.6)	0.31
Reveal	5.3 (0.0)	6.9	5.1 (0.0)	3.7	10.4 (0.0)	0.50
Bool2	5.3 (0.0)	7.6	4.0 (0.0)	3.7	9.7 (0.0)	0.46
MRBN	2.1 (0.77)	3.0	9.0 (1.6)	6.9	11.1 (1.9)	0.53
Arkin	1.2 (0.07)	3.2	8.8 (0.03)	7.8	10.0 (0.07)	0.48
Weaver	5.6 (0.03)	4.7	7.3 (0.02)	3.4	12.9 (0.01)	0.61
D'Haeseleer	7.8 (0.03)	11.2	0.8 (0.03)	1.2	8.6 (0.08)	0.41

Table 5. Aggregated samples; $S = 20$ and $R = 100$; linear transfer functions were used

The results for the Spellman *et al.* (1998) data are reported in Table 6. These data were restricted to the 11 cell cycle genes that appear in the diagram published by Lee *et al.* (2002). We

applied the PC and CCD algorithms to the data as though it were equilibrium data, using the 11 genes as variables, implicitly violating i.i.d. sampling assumptions of these algorithms. There are 66 possible regulatory relationships, ignoring direction of regulation, including autoregulation. Random assignment would thus imply 33 errors (and an error rate of 0.5).

	False Pos	Correct Pos	False Neg	Correct Neg	Total errors	Error rate
PC05	5	3	26	32	31	0.47
CCD05	5	3	26	32	31	0.47
Reveal	16	13	16	21	32	0.48
Bool2	2	1	28	35	30	0.45
MRBN	18	6	23	19	41	0.62
Arkin	3	2	27	34	30	0.45
Weaver	12	18	11	25	23	0.35
D'Haeseleer	2	1	28	35	30	0.45

Table 6. *S. Cerevisiae* data from Spellman et al. (2002)

These simulations clearly demonstrate that no confirmation of an algorithm for obtaining regulatory structure from expression data can be rationally justified by results with simulated data unless the data generating model is non-linear, with feedback, and the variable values are aggregated over simulated individual cells. Selective comparisons with independent wet-laboratory results do not suffice either. Among the tests reported here, Tables 2 and 3 describe the most realistic simulations, and the best tests, of the algorithms considered. Even so, in several respects the inference problems posed for those simulated datasets are easier than with real data: the correct time sampling frequency is known; all replications are with the same simulated metabolism and the same time sampling; there are no missing values; and the variables are aggregated over only 20 units (the larger the number of units of aggregation, the smaller the correlation among the aggregated variables).

The implementations of Reveal and Bool2 limited them to three regulators per gene. For those yeast genes actually with three or fewer regulators in the Lee *et al.* (2002) model, the results for these algorithms were almost always at chance, indicating the restriction to three regulators was inessential to their performance. One run of Bool2 that allowed for up to four regulators was attempted for the yeast data; the program ran for about 8 hours (over 200 times as long as the three regulator case) and returned the null model (no estimated regulatory relationships). On the simulation data, the Reveal, Bool2, Weaver, and D'Haeseleer algorithms proved useless; the remaining algorithms proved to be of some slight utility.

Considering both positive and negative errors, the Reveal, Bool2, MRBN and Arkin algorithms performed essentially at chance in all experiments: they are equivalent to flipping a coin to decide adjacencies. For non-linear simulated data, Weaver's algorithm is equivalent to saying "yes" to every adjacency; for linear data one would do better to use an inverted Weaver algorithm: say "yes" when it says "no." The D'Haeseleer algorithm is better than chance only for linear data, where it approximates saying "yes" in almost all cases. The PC and CCD are a little better than chance in all experiments and considerably better than chance with linear data. The Meek/Spirtes hybrid algorithm nearly dominates for total error rates on simulated data, and shows the theoretically expected increase in false positives with aggregated non-linear data. None of the algorithms improved with sample size increases up to 100. If we consider only the ratio of

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

correctly predicted positives to predicted positives, and the most realistic simulations (Tables 2 and 3), the PC, CCD, Meek/Spirtes, MRBN and Arkin algorithms all do slightly better than merely saying “yes” in all cases, varying in Table 4 from .62 to .64 as against the constant “yes” ratio of .57. The MRBN and Arkin algorithms purchase the slight improvement at the cost of missing most of the true positives.

These results tend to confirm the theoretical arguments against the reliability of machine learning algorithms for estimating gene regulation networks from microarray measurements of expression levels. The Meek/Spirtes algorithm, which does notably better than chance or constant “yes” responses on non-linear, non-aggregated data in Table 1, falls to the constant “yes” error rate when the variables are aggregated. The linear regression procedures overfit with data from a linear feedback system. It is conceivable that other counting principles would decrease false positives for PC, CCD and Spirtes/Meek and perhaps other algorithms in Tables 2 and 3, rendering them more useful, but we have not explored the possibilities. It would be preferable to have each algorithm run by its authors on common, well-specified, realistic simulation data from structures kept secret from those executing the algorithms and with explicit, pre-specified principles for counting errors, but such cooperative tests seem unlikely while relevant authors make neither their algorithms nor implementations publicly available.

COMPLEXITY WITHOUT CONDITIONAL INDEPENDENCE

The previous sections considered learning regulatory network structure from conditional independence information in microarray measurements. Network structure learning can instead be based on comparisons between the expression levels of various genes in (i) “wild type” cells that are not experimentally manipulated; and (ii) strains in which the expression levels of various genes have been suppressed or enhanced. This strategy essentially follows the logic of standard causal inference from experimental interventions and controls, though supplemented with algorithms that attempt to extract maximal information from the data. If we manipulate gene G_1 and the expression level of G_2 changes, then G_1 must a cause—direct or indirect—of G_2 . Moreover, this strategy does not use conditional independence information, and so the theoretical and experimental results about aggregation are irrelevant.

In principle, N experiments, each manipulating one of N genes, would suffice to identify the entire network if it is acyclic (without feedback), all effects are transitive, and no gene is both a direct and indirect regulator of any other gene. If $G(i)$ is the set of genes whose expression levels are altered when the expression level of gene i is experimentally randomized, then $G_k \rightarrow G_j$ if and only if $G(j) \subset G(k)$ and there is no r such that $G(j) \subset G(r) \subset G(k)$. If we can manipulate multiple genes at a time and trust conditional independence information, then the number of experiments can be reduced to around $\log_2(N)$ (Eberhardt, 2007), but for reasons noted above these methods are not applicable to real-world gene regulation data with, e.g., feedback (Frenster and Hovsepian, 2002).

This section considers the number of experiments required when cyclic network graphs are possible, and we set aside the statistical difficulties in determining differential expression (though we return to those issues at the end of this section). We also consider here the practical question of which experiment should be performed next, given the current state of one’s knowledge. In contrast to suggestions in some of the literature (e.g., Onami *et al.*, 2001; Ideker *et al.*, 2000), we argue that experimental manipulations do not permit efficient search for the true regulatory network. We continue to use graphical model representations of regulatory networks, but now allow cyclic graphs. We understand edges in terms of idealized experimental manipulations: $X \rightarrow$

Y in a network of genes \mathbf{V} if and only if there are experimentally producible values of X , $x_1 \neq x_2$, such that the expression level of Y differs for x_1 and x_2 when all other genes are held fixed. We cannot in practice hold most gene expression levels fixed at some value, even the “wild type” level; we have only techniques to suppress or overexpress a gene. We therefore focus on experiments in which a subset of the genes are experimentally suppressed or overexpressed while one measures the expression levels of the other genes. These experimental limitations may prevent us from learning the precise structure of certain networks (e.g., if an indirect influence is only detectable by (impossibly) holding the mediating gene fixed at its wild type level).

This edge semantics implies an obvious inference principle: Infer $G \rightarrow H$ if and only if there are experiments E and E^* such that (i) G is manipulated to different values in E and E^* ; (ii) H is not manipulated in E and E^* ; (iii) H 's expression level differs between E and E^* ; and (iv) E and E^* do not differ in their treatment of any other variable. This inference principle is a precise statement of an obvious idea: gene i regulates gene j just when a change in the experimental manipulation of gene i (while not changing anything else in the system) leads to a change in the expression level of gene j . Crucially, this inference principle depends only on (significant) differences in expression levels, rather than on conditional independencies; as such, it is not subject to problems due to aggregation.

The worst-case complexity for number of experiments arises when none of the genes in the network regulate any others. An inference that G definitely does not regulate H requires finding that H has the same expression level for the three different experimental manipulations of G —wild type, suppressed, and overexpressed—for every combination of the three possible treatments for each of the other variables. Manipulation of other genes is required because of the possibility that G is a redundant regulator of H : some other gene L also regulates H , and so the influence of G is noticeable only for particular settings of L . G and L might alternately have a complex interaction in regulating H . Because of these possibilities, we cannot simply manipulate G and look for a change in H . Exclusion of $G \rightarrow H$ from the network thus requires $3 \times 3^{n-2}$ distinct experiments. If no genes regulate any others, then one must conduct all of these experiments for every ordered pair of genes, and so $n \times (n - 1) \times 3 \times 3^{n-2}$ distinct experiments will be required in all.

Any reliable inference algorithm must thus use information from exponentially many different experiments in the worst case, and so any reliable inference algorithm must itself have exponential complexity in the worst case. For $n = 9$ (as in the empirical data we consider below), reliable inference requires 472,392 experiments in the worst case. The worst-case number of experiments is required only if G does not regulate H ; if G actually does regulate H , then that can be reliably discovered in as few as two experiments. The expected and real-world computational complexity of algorithms based on this inference principle will almost certainly be much less than the worst-case bound. In general, the algorithmic complexity decreases as the regulatory network density increases (i.e., as the number of edges goes up).

A different complexity analysis focuses on the number of networks that are consistent with some set of experiments, as well as the number of consistent, minimal networks. Suppose we have m distinct experimental conditions, each repeated l times, in which we measure the expression levels of n genes. Experimental conditions may differ in the genes that are suppressed or overexpressed, in various environmental conditions such as nutrient levels, or both. Let m_{ij} denote the mean expression level of gene i in experimental condition j , typically after normalizing distributions across conditions. We are most interested in a gene's expression level being different between two different experimental conditions, and so we assume that (using some

simultaneous hypothesis test) we obtain a statistical decision about whether $m_{ij} = m_{ik}$ for each gene i and all pairs of experimental conditions j and k in which gene i is not directly manipulated. The j, k pairs for which $m_{ij} \neq m_{ik}$ are the findings that must be explained; the search problem is to find graphs that explain all of the observed expression level changes, and in the case of minimal networks, only the observed changes.

Formally, we construct a three-dimensional matrix A of size [gene & exogenous condition] \times experimental condition \times experimental condition. For each gene i :

- $a_{ijk} = M$, if either gene i is the target of an experimental manipulation in only one of j and k , or if it is manipulated in different ways in j and k ;
- $a_{ijk} = 1$, if gene i is not manipulated in j and k , and the statistical decision is that $m_{ij} \neq m_{ik}$ (i.e., gene i has significantly different expression in the two conditions); and
- $a_{ijk} = 0$, otherwise (i.e., either gene i is experimentally manipulated in the same way in both conditions, or there is no significant difference in mean expression level).

For each exogenously controlled experimental condition h (e.g., temperature, nutrient level):

- $a_{hjk} = 1$, if h 's value differs between conditions j and k ; and
- $a_{hjk} = 0$, otherwise.

Define a graph G to be *consistent* with a set of experimental results A if and only if: (i) G does not contain any edges incompatible with A (in a sense defined below); and (ii) for all $a_{ijk} = 1$, there exists a gene q such that there is a directed path in G from q to i , and $a_{qjk} =$ either 1 or M . The second condition ensures that G can explain every significant difference in expression levels. The first condition ensures that G contains no impossible (relative to the data) edges. More precisely, we define a $G \rightarrow H$ edge to be *incompatible* with A if and only if: for all possible combinations of experimental manipulations (including no manipulation) of all genes except G and H , the expression level of H does not change regardless of the state of G . Note that an edge is only definitely incompatible with A if we perform every experiment in which H is unmanipulated (i.e., 3^{n-1} experiments); it is quite difficult to definitively rule out regulatory dependencies when we allow redundant pathways, cycles, and nonlinearities.

We are often most interested in the *minimally consistent* graphs for A : those that are consistent with A , but not consistent if any edge is removed. The *IG (Initial Graphs)* algorithm finds such graphs:

- 1) For each gene i and all j, k such that $a_{ijk} = 1$, let L_{ijk} be the set of genes and exogenous factors l such that $a_{ljk} =$ either 1 or M .
- 2) For each gene i , determine C_i : the set of minimal covering sets for all non-empty L_{ijk} .
- 3) Construct the collection of directed graphs, \mathbf{G} , consisting of every possible graph that can be formed by choosing, for each i , some $c_i \in C_i$ and then making all factors in c_i into parents of i .
- 4) For each pair of experimental conditions j, k , let \mathbf{I} be the set of factors such that $a_{ijk} = M$ (or 1, in the case of exogenous factors). For each gene r such that $a_{rjk} = 1$, and for all $G \in \mathbf{G}$, if there is no directed path in G from a member of \mathbf{I} to r , then replace G with all extensions of G that add a directed edge from a member of \mathbf{I} to r .
- 5) Return \mathbf{G} (henceforth, called **InitialGraphs**).

In plain language, steps 1 and 2 determine the “minimal” explanations for all differences in expression level. Step 3 then constructs the graphs corresponding to all possible combinations of minimal explanations. Finally, step 4 ensures that all of the explanations ultimately ground out in an experimental manipulation. Note that this algorithm can output cyclic graphs.

As an example, suppose Table 7 gives example expression data, where ‘ w_{t_i} ’ indicates the wild type expression level of gene i . Suppose further that neither ε nor δ is itself a significant change (from w_{t_3}), but $\varepsilon + \delta$ is a significant change in expression level. In that case, the corresponding A matrix is given in Table 8.

	Gene 1	Gene 2	Gene 3
Experiment 1	w_{t_1}	w_{t_2}	w_{t_3}
Experiment 2	suppressed	w_{t_2}	$w_{t_3} + \varepsilon$
Experiment 3	w_{t_1}	suppressed	$w_{t_3} - \delta$

Table 7. Example expression data in three experiments

$a_{112} = M$	$a_{212} = 0$	$a_{312} = 0$
$a_{113} = 0$	$a_{213} = M$	$a_{313} = 0$
$a_{123} = M$	$a_{223} = M$	$a_{323} = 1$

Table 8. Significant difference matrix for example experiments

The expression levels of both genes 1 and 2 change significantly only between experimental conditions with different manipulations of that gene. Therefore, for those two genes, there are no sets to be covered; C_1 and C_2 are both the empty set. For gene 3, there is only a significant change in expression level between experiments 2 and 3. $L_{323} = \{\text{gene 1, gene 2}\}$ since both change, and so the minimal covering sets for gene 3 are $\{\text{gene 1}\}$ and $\{\text{gene 2}\}$. The IG algorithm thus outputs two different graphs: $1 \rightarrow 3 \rightarrow 2$; and $1 \rightarrow 3 \leftarrow 2$.

We can prove the following theorem about the contents of IG algorithm’s output:

Theorem 4 (IG consistency): Every graph in **InitialGraphs** is consistent with A , and **InitialGraphs** contains all graphs that are minimally consistent with A .

InitialGraphs typically does not include all of the consistent graphs. Provably, if G is consistent with A and $i \rightarrow j$ is not incompatible with A , then the graph formed by adding $i \rightarrow j$ to G is consistent with A . If our primary interest is in the *number* of consistent regulatory networks, then we can add together the number of supergraphs of each $G \in \mathbf{InitialGraphs}$, and then subtract out the graphs that are doubly-counted. This procedure scales up poorly, however, as it requires, for m initial graphs, calculating the size of $2^m - 1$ sets (i.e., supergraphs of all initial graphs, and all of the different overlaps). We can, however, use this strategy to compute a lower bound on the number of consistent graphs, since the number of supergraphs of a subset of **InitialGraphs** is necessarily less than or equal to the number of supergraphs of all of **InitialGraphs**. We can also determine the number of supergraphs of one graph in **InitialGraphs**, then the number of supergraphs of two graphs, then... until we include all of **InitialGraphs**. We can optimize the calculation by starting with the sparsest graph(s), since they will contribute the largest terms. Finally, since the computed lower bound increases (weakly) monotonically at each stage, we can run the procedure and stop at any time to get a lower bound.

The IG algorithm can also be used as the basis for a procedure to select which experiment to perform next. An algorithm to find the globally optimal next experiment would need to consider all possible sequences of unperformed experiments, the possible outcomes in each experiment in each sequence, and the number of consistent graphs for each possible outcome after each experiment in a possible sequence. This computation is hopelessly intractable; if we have conducted L experiments, then we must compute the exact number of consistent graphs at every branch-point and leaf in an exponentially branching tree of depth $(3^n - L)$ corresponding to a

sequence of experiments, where there are $(3^n - L)!$ many sequences/trees. Alternatively, we could use the IG algorithm in a heuristic procedure to select the best next experiment. For some set \mathbf{E} of possible experiments, compute for each experiment E the expected number of consistent graphs over all possible outcomes of E , and then perform the experiment E^* that minimizes this number. One can clearly add a probability distribution over outcome likelihood for an experiment, if that prior knowledge is available. This procedure appears more promising, but still faces significant challenges. It is a greedy search procedure, and so can lead to sub-optimal sequences of experiments. It is also still quite computationally complex: there will typically be exponentially many unperformed experiments, and we must compute the exact number of consistent networks (i.e., conduct an exponential calculation) for every possible outcome of every $E \in \mathbf{E}$.

We have given theoretical reasons to question whether the basic inference principle underlying manipulation experiments can be generalized to a reliable search procedure. These concerns might fail to be an issue for realistic networks, however, and so we now consider a real-world case. Ideker *et al.* (2001) conducted a series of experiments on the galactose metabolism cycle in yeast (*Saccharomyces cerevisiae*). They used microarrays to measure expression levels of 5000 different genes, but focused on nine genes that had previously been identified as important in this cycle. They performed ten experiments (wild type measurements, plus single knockouts of each gene) in two environments (presence and absence of galactose). We focus here on the galactose present case.

Ideker *et al.* (2001) performed four replications of each experimental condition, with appropriate instrumental counter-balancing across replications, in order to improve the power of statistical tests. There were nonetheless many statistical challenges to constructing the \mathbf{A} matrix that encodes significant differences in expression level between experimental conditions. The most significant issues were correlated errors in the measurements, and the large number of simultaneous statistical tests required for this 9-gene/10-experiment setup. For reasons of space, we do not go into details here; a full statement of the statistical procedures that we used can be found in Danks *et al.* (2003).

Despite these challenges, the Ideker *et al.* (2001) data are close-to-ideal for the IG algorithm. We have expression levels for all genes in all ten experiments, and so can carry out all of the required pairwise comparisons. Automated search is also clearly required for these data, since there are $2^{72} \approx 4 \times 10^{21}$ possible regulatory networks over these nine genes. The output of the first three steps of the IG algorithm (i.e., the minimal covering sets for each gene) are shown in Table 9. Each column gives the minimal covering sets for that particular gene, where each row within a column is a different minimal covering set.

<i>gal1</i>	<i>gal2</i>	<i>gal3</i>	<i>gal4</i>	<i>gal5</i>	<i>gal6</i>	<i>gal7</i>	<i>gal10</i>	<i>gal80</i>
10	7	7	1	1,3	1,2	10	1,6	4,5
2,6,7	10	10	3	1,4	2,7	1,3	1,2,80	4,6
2,7,80		1,2	5	3,6	2,10	1,2	1,2,5	4,10
		1,6	7	2,4,6	5,7	1,6	2,3,6	5,7
		1,80	10	4,6,7	5,10	1,80	2,7,80	6,7
		4,6		4,6,10	1,3,5		2,6,7	7,10
		2,5,6			1,5,80		4,6,7	
		5,6,80			2,3,5		2,3,5,80	
					2,5,80			

Table 9. Minimal covering sets for galactose genes in Ideker, *et al.* (2001) data

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

This table describes 3,110,400 different cyclic and acyclic graphs corresponding to all possible ways of choosing parents (i.e., minimal covers) for each gene. After step (4) of the IG algorithm—checking whether every expression level change is explained in every graph—the IG procedure returns a set of 3,480,675 graphs.

The output of the IG algorithm includes all networks that are minimally consistent with **A**, but not necessarily all consistent graphs. The sparsest graph in **InitialGraphs** contains fourteen edges, and no edges are inconsistent with the data. A lower bound (on the total number of consistent graphs) calculated using only this sparsest graph is approximately 2×10^{17} networks. The actual number of consistent graphs is almost certainly larger. These ten experiments have thus reduced the possibility space by (at most) four orders of magnitude, and probably much less.

This section began with the hope that differences in expression levels between experimental conditions could provide the basis for reliable search procedures. The theoretical and empirical results argue, however, that such data actually provide relatively little information: there are typically many different potential regulators for some expression level difference, and too many experiments are required to rule out a regulatory connection. At the same time, gene manipulation experiments can be expected to remain an important confirmatory method for testing a specific hypothesis. They are ineffective for search, but quite powerful for targeted hypothesis testing.

CONCLUSION

Machine learning methods offered the promise of a shortcut to discovering genetic regulatory networks. The promise has so far proved false, for reasons we have described. Machine learning methods—in particular, automated search for graphical causal models—are applicable to many other genomics problems, and potentially even to gene regulation problems when data are available at the individual cellular level. When our data come from aggregations of cells, however, then sophisticated machine learning methods are actually penalized, since the conditional independence relations on which they depend do not hold at the aggregate level. Simpler methods that do not exploit conditional independence relations (such as the IG algorithm in the previous section) are computationally intractable for realistic scenarios. There are of course multiple cases that purport to show successful machine learning from microarray data; these instances almost all (i) use unrealistic single-cell (simulated) data; (ii) focus on cases in which the ground truth is not known and so performance cannot be evaluated; or (iii) report only isolated successfully discovered regulatory connections, rather than statistics about overall algorithm performance. We thus conclude on a pessimistic note: despite the hopes of many (including us at a prior time), standard structure learning algorithms cannot be fruitfully applied to microarray data; rather, successful machine learning for genetic regulatory networks will depend on statistical, algorithmic, and experimental advances that are highly tuned to the challenges of this particular domain, and that largely remain to be done.

REFERENCES

- Akutsu, T., Miyano, S., & Kuhara, S. (2000). Algorithms for inferring qualitative models of biological networks. *Pacific Symposium on Biocomputing*, 5, 290-301.
- Arkin, A., Shen, P., & Ross, J. (1997). A test case of correlation metric construction of a reaction pathway from measurements. *Science*, 277, 1275-1279.

(in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57, 289-300.

Brown, C. T., Rust, A. G., Clarke, P. J. C., Pan, Z., Schilstra, M. J., Buyscher, T. D., Griffin, G., Wold, B. J., Cameron, R. A., Davidson, E. H., & Bolouri, H. (2002). New computational approaches for analysis of cis-regulatory networks. *Developmental Biology*, 246, 86-102.

Chu, T. (2003). Learning from SAGE data. Unpublished doctoral dissertation, Carnegie Mellon University.

Chu, T. (2004). Limitations of statistical learning from gene expression data. *Interface 2004: Computational Biology and Bioinformatics*.

Chu, T., Glymour, C., Scheines, R., & Spirtes, P. (2003). A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays. *Bioinformatics*, 19, 1147-1152.

Danks D. & Glymour, C. (2002). Linearity properties of Bayes nets with binary variables. In J. Breese & D. Koller (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 17th conference (UAI-2001)* (pp. 98-104). San Francisco: Morgan Kaufmann.

Davidson, E., Rast, J., Oliveri, P., Ransick, A., Calestani, C., Yuh, C., Minokawa, T., Amore, G., Hinman, V., Arenas-Mena, C., Otim, O., Brown, C., Livi, C., Lee, P., Revilla, R., Rust, A., Pan, Z., Schilstra, M., Clarke, P., Arnone, M., Rowen, L., Cameron, R., McClay, D., Hood, L., & Bolouri, H. (2002). A genomic regulatory network for development. *Science*, 295, 1669-1678.

D'haeseleer, P. (2000) Reconstructing gene networks from large scale gene expression data. Unpublished doctoral dissertation, University of New Mexico.

D'haeseleer, P., Liang, S., & Somogyi, R. (2000). Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics*, 16, 707-726.

D'haeseleer, P., Wen, X., Fuhrman, S., & Somogyi, R. (1999). Linear modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing*, 4, 41-52.

Eberhardt, F. (2007). Causation and intervention. Unpublished doctoral dissertation, Carnegie Mellon University.

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, 297, 1183-1186.

Frenster, J. H., & Hovsepian, J. A. (2002). RNA feedback mechanisms during eukaryotic gene regulation. In *Northwest symposium on systems biology* (p. 15).

Friedman, N., Nachman, I., & Pe'er, D. (1999). Learning Bayesian network structure from massive datasets: The 'sparse candidate' algorithm. In K. Laskey & H. Prade (Eds.), *Proceedings of the 15th international conference on uncertainty in artificial intelligence* (pp. 206-215). San Francisco, CA: Morgan Kaufmann.

Friedman, N., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Recomb 2000*, Tokyo.

- (in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.
- Genovese, C., & Wasserman, L. (2001). False discovery rates (Technical report 762). Carnegie Mellon University: Department of Statistics.
- Hartemink, A. (2001). Principled search for gene regulation. Unpublished doctoral dissertation, Harvard University.
- Hartemink, A. (2006). Bayesian networks and informative priors: Transcriptional regulatory network models. In Do, K.-A., Müller, P., & Vannucci, M. (Eds.), *Bayesian inference for gene expression and proteomics* (pp. 401-424). Cambridge: Cambridge University Press.
- Hashimoto, R. F., Kim, S., Shmulevich, I., Zhang, W., Bittner, M. L., & Dougherty, E. R. (2004). Growing genetic regulatory networks from seed genes. *Bioinformatics*, *20*, 1241-1247.
- Ideker, T. E., Thorsson, V., & Karp, R. M. (2000). Discovery of regulatory interactions through perturbation: Inference and experimental design. *Pacific Symposium on Biocomputing*.
- Ideker, T., Thorsson, V., Ranish, J., Christmas, R., Buhler, J., Eng, J., Bumgarner, R., Goodlett, D., Aebersold, R., & Hood, L. (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, *292*, 929-934.
- Lee, T., Rinaldi, N., Robert, F., Odom, D., Bar-Joseph, Z., Gerber, G., Hannett, N., Harbison, C., Thompson, C., Simon, I., Zeitlinger, J., Jennings, E., Murray, H., Gordon, D. B., Ren, B., Wyrick, J., Tagne, J., Volkert, T., Fraenkel, E., Gifford, D., & Young, R. (2002). Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, *298*, 799-804.
- Levsky, J. M., Shenoy, S. M., Pezo, R. C., & Singer, R. H. (2002). Single-cell gene expression profiling. *Science*, *297*, 836-840.
- Liang, S., Fuhrman, S., & Somogyi, R. (1998). Reveal: a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, *3*, 18-29.
- Onami, S., Kyoda, K. M., Morohashi, M., & Kitano, H. (2001). The DBRF method for inferring a gene network from large-scale steady-state gene expression data. In H. Kitano (Ed.), *Foundations of systems biology* (pp. 59-75). Cambridge, Mass.: The MIT Press.
- Pe'er, D., & Hartemink, A. (2004). Single-cell gene expression analysis: Implications for neurodegenerative and neuropsychiatric disorders. *Neurochemical Research*, *29*, 1053-1064.
- Pe'er, D., Regev, A., & Tanay, A. (2002). MinReg: Inferring an active regulator set. In *Proceedings of the tenth international conference on intelligent systems for molecular biology (ISMB)*.
- Pe'er, D., Tanay, A., & Regev, A. (2006). MinReg: A scalable algorithm for learning parsimonious regulatory networks in yeast and mammals. *Journal of Machine Learning Research*, *7*, 167-189.
- Richardson, T. (1996). A discovery algorithm for directed cyclic graphs. In *Proceedings of the 12th international conference on uncertainty in artificial intelligence* (pp. 454-461).
- Rosenfeld, N., Young, J. W., Alon, U., Swain, P. S., & Elowitz, M. B. (2005). Gene regulation at the single-cell level. *Science*, *307*, 1962-1965.
- Schilstra, M. (2002). NetBuilder software. <http://strc.herts.ac.uk/bio/maria/>.

- (in press). In S. Das, D. Caragea, W. H. Hsu, & S. M. Welch (Eds.), *Computational methodologies in gene regulatory networks*. Hershey, PA: IGI Global Publishing.
- Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18, 261-274.
- Shrager, J., Langley, P., & Pohorille, A. (2002). Guiding revision of regulatory models with expression data. *Pacific Symposium on Biocomputing*, 7, 486-497.
- Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, B., & Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9, 3273-3297.
- Spirtes, P., Glymour, C., & Scheines, R. (2001). *Causation, Prediction and Search*. 2nd edition, Cambridge, MA: The MIT Press.
- Spirtes P., & Meek, C. (1995). Learning Bayesian networks with discrete variables from data. In U. M. Fayyad & R. Uthurusamy (Eds.), *Proceedings of the first international conference on knowledge discovery and data mining* (pp. 294-299). San Jose, CA: AAAI Press.
- van Someren, E. P., Wessels L. F. A., & Reinders, M. J. T. (2000). Linear modeling of genetic networks from experimental data. In *Proceedings of the eighth international conference on intelligent systems for molecular biology* (pp. 355-366).
- Weaver, D. C., Workman, C. T., & Stormo, G. D. (1999). Modeling regulatory networks with weight matrices. *Pacific Symposium on Biocomputing*, 4, 112-123.
- Yoo, C., Thorsson, V., & Cooper, G. F. (2002). Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. *Pacific Symposium on Biocomputing*, 7, 498-509.
- Yuh, C., Bolouri, H., & Davidson, E. (1998). Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene. *Science*, 279, 1896-1902.