The Computational and Experimental Complexity of Gene Perturbations for Regulatory Network Search

David Danks
Institute for Human & Machine Cognition
University of West Florida
Pensacola, FL 32501
ddanks@ai.uwf.edu

Clark Glymour
IHMC and
University of West Florida
Pensacola, FL 32501
cg09@andrew.cmu.edu

Peter Spirtes
Department of Philosophy
Carnegie-Mellon University
Pittsburgh, PA 15213
ps7z@andrew.cmu.edu

Abstract

Various algorithms have been proposed for learning (partial) genetic regulatory networks through systematic measurements of differential expression in wild type versus strains in which expression of specific genes has been suppressed or enhanced, as well as for determining the most informative next experiment in a sequence. While the behavior of these algorithms has been investigated for toy examples, the full computational complexity of the problem has not received sufficient attention. We show that finding the true regulatory network requires (in the worst-case) exponentially many experiments (in the number of genes). Perhaps more importantly, we provide an algorithm for determining the set of regulatory networks consistent with the observed data. We then show that this algorithm is infeasible for realistic data (specifically, nine genes and ten experiments). This infeasibility is not due to an algorithmic flaw, but rather to the fact that there are far too many networks consistent with the data (10^{18} in the provided example). We conclude that gene perturbation experiments are useful in confirming regulatory network models discovered by other techniques, but not a feasible search strategy.

1 Introduction

Techniques for the simultaneous measurement of mRNA transcripts from thousands of genes has led to the development of a wide range of machine learning techniques for searching for genetic regulatory networks in an automated fashion. In a large subset, the learning is based on comparisons between the gene expression in "wild type" cells and that in strains in which expression of particular genes has been deliberately 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.

Our primary interest is in determining how many gene perturbation experiments are required to determine the network of regulatory relations among any given set of genes, ignoring questions of uncertainty in statistical decisions. Secondarily, we are interested in whether it is feasible to compute which further experiments will be most informative. And, finally, since the sample sizes (number of expression measurements per gene) in such experiments are typically small, we are concerned with the stability of statistical decisions about differential expression given different statistical tests.

The answers we find are far more pessimistic than has previously been suggested in the literature (e.g., [Onami *et al.*, 2001; Ideker *et al.*, 2000]). We show that, while perturbation experiments can eventually eliminate possible regulatory relations, they do not efficiently eliminate them. We give an anytime algorithm for computing weakly monotonically increasing lower bounds on the number of alternative network explanations for the results of any set of gene perturbation experiments. The lower bound is typically astronomical. We illustrate the point by computing a lower bound—10^{18}—on the number of networks for 9 genes that are consistent with a recent series of gene perturbation experiments [Ideker *et al.*, 2001]. Finally, we argue that the computation of the most informative experiments can only be heuristic.

2 Graphical Representation

Regulatory networks have often been represented as directed acyclic graphs. However, since feedback is ubiquitous in gene regulation, either a time series or a directed cyclic graph representation is more appropriate, and we will use the latter. An edge $X \rightarrow Y$ in a regulatory network indicates that gene X directly regulates gene Y (relative to the other variables in the network). We can put the idea more precisely in terms of idealized experiments: X directly regulates Y relative to variables $\mathbf{V}$ iff there are two distinct experimentally producible values, x_1, x_2 of X, such that, if the expression levels of all genes other than X, Y ($\mathbf{V}$ \

$\{X,Y\}$) are held fixed, the expression levels of Y are distinct for the two values of X.

In practice, we do not have experimental techniques to hold gene expression levels at arbitrary values, only techniques to suppress or overexpress a gene or genes. We cannot even force a gene to have its wild-type expression level if we experimentally suppress or overexpress other genes. We therefore focus on the problem of discovering the networks that can be revealed by experiments that suppress or overexpress specific genes, singly or in combination, while measuring expression levels of other genes. Networks discovered in this way necessarily may be incomplete; for example, if gene X regulates gene Z only when gene Y has a wild-type value, and under or over expression of X also drives Y away from its wild-type level, the influence of X on Z may not be discovered.

3 Counting Experiments

Suppose we have n different genes. In the mathematically worst case (when no gene regulates any other gene), we must perform $n \times (n–1) \times 3 \times 3^{n–2}$ different experiments to learn the true network—so much as it can in principle be learned from such experiments. The first two terms measure the number of different ordered pairs of genes we must consider. The third term measures the different possible interventions on the potential regulator gene (left alone, knocked out, overexpressed). The final term measures the number of different possible experimental "settings" for the other variables in our system (left alone, knocked out, overexpressed). If future experimental technology enables us to set variables at more than two levels, the last two terms will increase exponentially.

An immediate consequence is that any generally correct algorithm for determining the graphical models consistent with a collection of experimental results must, in the worst-case, consider exponentially many different experiments. Thus, any such algorithm must have exponential worst-case computational complexity, since (at the very least) every experimental result must be read once. Even for a relatively small case such as the yeast data discussed below in which $n = 9$, there are (worst-case) 472,392 experiments to be performed.

Of course, expected or real-world cases are often easier than the worst case. We must perform the full $2 \times 3 \times 3^{n–2}$ experiments for a particular pair of genes only if neither actually regulates the other. If we can determine that X directly regulates Y, then no more experiments are needed for that ordered gene pair (at least if network topology is all that concerns us). The computational complexity thus decreases as the density of the regulatory network increases. Three questions about complexity are suggested by these considerations: Given a collection of experimental results, (i) what is the class of network structures that are consistent with the data; (ii) what is its cardinality; and, since that cardinality is apt to be very large, (iii) is there a way to generate a smaller class containing the "minimal" graphs, at least one of which must be true in view of the data? To answer these questions we need a formal representation of the data and of the notion of a network structure being "consistent" with the data.

4 Formal Representation of Data

Suppose the expression of n genes is measured in m experimental conditions, with l repetitions of each experimental condition. Experimental conditions may differ in the genes that are deliberately enhanced or suppressed, in environmental factors such as temperature or nutrient, or both. After normalizing the distributions of each gene in each condition, we obtain an estimate of the mean expression m_{ij} of gene i in experimental condition j. We will assume that, using some simultaneous hypothesis test, we obtain a test of the hypothesis that $m_{ij} = m_{ik}$, for each gene i, and each pair of distinct experiments j and k, in neither of which gene i is directly manipulated. We construct a ([gene & exogenous factor] × experiment × experiment) symmetric matrix $\boldsymbol{A}$ and set:

a_{ijk} = M, if gene i is the target of an experimental manipulation in at least one of experiment j or k, and it is not the same manipulation in both experiments;

$a_{ijk} = 1$, if the hypothesis is rejected (i.e., expression of gene i is significantly different in j and k);

$a_{ijk} = 0$, otherwise.

For each exogenously controlled environmental factor h we set:

$a_{hjk} = 1$, if h's value differs in experiments j and k

$a_{hjk} = 0$, otherwise.

We require a definition of the conditions under which an edge is ruled out by the observed data. Specifically, an edge $X \rightarrow Y$ is *incompatible with observed data* $\boldsymbol{A}$ if and only if for all combinations of possible intervention states (including no intervention) for the genes except X and Y, the expression level of Y does not change regardless of the intervention state of X. In other words, for every way of setting the values of the other genes (including not doing anything at all), genome perturbations in X never lead to variations in Y. Note that we must in fact perform every possible experiment that leaves Y unmanipulated (i.e., $3^{n–1}$ experiments) to determine whether an $X \rightarrow Y$ edge is incompatible. In practice we will almost never have sufficient experiments to declare an edge incompatible with the data. Nevertheless, it is a useful definition, if only because it emphasizes just how difficult it is to rule out regulatory dependencies.

Now define a graph G to be *consistent with a set of experimental results* $\boldsymbol{A}$ if and only (i) G does not contain any edges incompatible with E, in the sense of "incompatible" just defined; and (ii) for all a_{ijk} in E such that $a_{ijk} = 1$, there exists a gene vertex X_q such that there is a directed path from X_q to X_i, and either $a_{qjk} = 1$ or M. In plain language, the first clause of the definition requires that G does not include any impossible (relative to the data) edges, and the second clause requires that every significant expression difference in the data can be explained by G.

5 Finding Minimally Consistent Graphs

Given the matrix $\boldsymbol{A}$, we define the *IG Algorithm (Initial Graphs)*:

(i) For each gene i, and for all j, k such that $a_{ijk} = 1$, let L_{ijk} be the set of l (genes and exogenous factors) such that $a_{ljk} =$ 1, or M. Given the L_{ijk}, determine C_i: the collection of minimal covering sets for these L_{ijk};

(ii) Construct the collection $\mathbf{G}$ of all possible directed graphs in which the parent nodes of each gene i form one of the minimal covering sets from C_i.

(iii) For each pair of experiments, j, k, form the set $\mathbf{I}$ of all i such that $a_{ijk} =$ M. For each r such that $a_{rjk} = 1$ and G in $\mathbf{G}$, if there is no directed path from a member of $\mathbf{I}$ to r, replace G (in $\mathbf{G}$) with all extensions of G that add a directed edge from some member of $\mathbf{I}$ to r.

(iv) return $\mathbf{G}$ (henceforth, **InitialGraphs**)

The algorithm allows the generation of cyclic graphs, which have a straightforward semantics: e.g., if X, Y form a two-cycle, then for some fixed values of other genes, a perturbation of X changes the expression of Y, and vice versa. Step (iii) of the algorithm is not redundant, and is required by the normal experimental method. However, this step can yield multiple copies of the same graph when applied to different members of **InitialGraphs**, and checking for these redundancies adds enormous computational complexity.

As an example, assume our expression data are:

Gene 1	Gene2	Gene 3
wt_1	wt_2	wt_3
deleted	wt_2	$wt_3 + \varepsilon$
wt_1	deleted	$wt_3 - \delta$

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

For both genes 1 and 2, the only significant changes occur between experiments in which the gene was the subject of an intervention. Therefore, for these two genes, there are no sets to be covered. For gene 3 there is a significant change in expression level (i.e., a '1') between experiments 2 and 3 (the last row of the matrix). Genes 1 and 2 both change significantly between experiments 2 and 3 (see the third row of the matrix) and therefore {gene 1} and {gene 2} are the minimal covering sets for gene 3. Hence, the IG procedure outputs two different graphs: "$G_1 \rightarrow G_3 \quad G_2$"; and "$G_1 \quad G_3 \leftarrow G_2$". Step 3 of the algorithm, path checking, produces no changes.

In our example, the "ground truth" was that genes 1 and 2 co-regulate gene 3, but the procedure tells us only that gene 1 regulates gene 3 *or* gene 2 regulates gene 3, where the "or" is inclusive. In general, the procedure leaves out many consistent graphs.

However, we say that a graph G is *minimally consistent* with $\boldsymbol{E}$ if and only if G is consistent with E, but if we remove any edge from G, then it is no longer consistent. We can prove that the output of the IG algorithm, while leaving out some consistent graphs, includes every minimally consistent graph.

> **Theorem 1**: (A) Every graph in **InitialGraphs** for $\boldsymbol{A}$ is consistent with $\boldsymbol{A}$, and (B) **InitialGraphs** contains all graphs minimally consistent with $\boldsymbol{A}$.

Note that **InitialGraphs** may contain graphs that are not minimal for the experimental data; an example is given, in another context, in the "Previous Work" section below.

We can extend the IG algorithm to include unobserved common causes. For each pair of gene nodes j, k correlated in some experimental condition, and for each graph G in **InitialGraphs**, if there is no directed path between j and k and there is no pair of directed paths from any third node to j and k, add a bidirected edge (i.e., '↔') to G between j and k. Unfortunately, the signal-to-noise ratio in gene expression measurements does not allow reasonable estimates of expression correlations [Chu, 2003].

6 The Exact Number of Consistent Graphs

Recall that we earlier defined the notion of an $X \rightarrow Y$ edge being incompatible with data $\boldsymbol{A}$. These edges are the only ones that never appear in a graph consistent with the data. More formally, we can prove the following theorem about the generation of consistent graphs.

> **Theorem 2**: If G is consistent with E and $X \rightarrow Y$ is not incompatible with E, then G^*, the supergraph of G formed by adding $X \rightarrow Y$, is consistent with E.

These two theorems enable us to determine the regulatory connections that appear in every consistent graph: X is a regulator of Y in every consistent graph if and only if X is in every minimal covering set of Y. We can also directly (though usually not feasibly) calculate the exact number of graphs consistent with the observed data: (i) Let $\mathbf{G}$ = **InitialGraphs**; (ii) iteratively add to $\mathbf{G}$ all graphs that can be formed by adding a compatible edge to some graph currently in $\mathbf{G}$; and (iii) calculate the cardinality of $\mathbf{G}$. This procedure is quite inefficient, since many duplicate graphs will be generated by this procedure, and checking for duplication is generally infeasible for large sets. A more modest aim is to calculate directly the exact number of consistent graphs.

The most obvious procedure would be to calculate the number of supergraphs of each graph in **InitialGraphs**, and then subtract out the overlap (i.e., those supergraphs that appear multiple times). It is easy to calculate the number of supergraphs of any one graph: if the graph has n nodes and k edges, and there are q incompatible edges, then there are $2^{n(n-1)-k-q}$ supergraphs. For every ordered pair of variables (X,Y), the edge can either be present or absent, unless it is already in the initial graph or is incompatible. Hence, the number of supergraphs is the number of Boolean combinations of edges whose presence or absence is not yet determined.

The size of the overlap between the supergraphs of G and H can also be straightforwardly computed: determine the graphical union of G and H (i.e., the graph with an $X \rightarrow Y$ edge iff at least one of G, H contains $X \rightarrow Y$), and calculate *its* number of supergraphs. But the general strategy is equivalent to calculating the cardinality of the union of multiple overlapping sets. The above algorithm thus requires, for m initial graphs, calculating the size of 2^m-1 sets (including the various overlaps), and so is clearly intractable for any large **InitialGraphs**.

This result suggests a more general conjecture: there is no procedure to calculate the exact number of graphs consistent with $\boldsymbol{E}$ in polynomial time. If this conjecture is true, as we suspect, then the most a tractable algorithm can do is calculate a lower bound. As we shall see, a lower bound will prove sufficient for assessing the feasibility of gene perturbation as a search strategy.

We can determine a lower bound on the number of graphs consistent with the observed data by calculating the number of supergraphs of some subset of **InitialGraphs**. We can thus calculate this lower bound for one graph, then two graphs, then… until we have calculated the lower "bound" for all of **InitialGraphs**, which is just the exact number of consistent graphs. This algorithm is more efficient than it might appear, since the terms used for the lower bound computation for n initial graphs can all be used for the computation for n+1 initial graphs. Moreover, by starting with the sparsest graph(s), we can quickly reach a reasonable bound. The computed lower bound increases (weakly) monotonically with each stage. We can stop the algorithm at any time after the first stage and receive useful output. Furthermore, there is some finite (though exponentially far away) time at which the algorithm will return the exact number of consistent graphs. If **InitialGraphs** is sufficiently small or we have sufficient time, then this algorithm will yield an exact answer.

7 Planning Experiments

Given the very small sample sizes in microarray and SAGE experiments, the confidence intervals for estimates of expression levels may be quite large, in which case the best experiment might be a replication of an earlier experiment to narrow this confidence interval. We have found that experiments using another method to detect expression of particular genes (e.g., Northern blots rather than microarrays) in order to confirm the effect of a treatment (e.g., applying a drug to a cell sample) are often the most urgent. We set these issues aside for the remainder of this section.

Given some set of graphs consistent with the observed data, we want to choose the best experiment to perform. We first focus on the problem of determining the globally optimal sequence of experiments (i.e., the sequence that most rapidly determines the correct regulatory network).

Suppose we have performed L experiments to this point. Then there are $R = 3^n - L$ experiments that have not been performed, and R! possible sequences. Systematic exploration of this sequence space is clearly intractable.

The computational complexity is actually worse than this factorial of an exponential. Suppose we could enumerate these possible sequences. For any particular sequence $I = i_1,\ldots,i_R$, we need to determine all of the possible outcomes after each experiment in the sequence. Let $W(E,F)$ to be the number of genes that receive *no* intervention in either experiment E or F. If S is the sequence of experiments already performed (both actually and hypothetically), then for some unique outcome of S, the number of possible outcomes after S+i is $O_i = \sum_{s \in S} 2^{W(i,s)}$. Calculating the total number of possible outcomes for a particular sequence I requires determining this branching structure (where there are O_i branches at the i-th stage).

For each of these possible outcome sequences, we need to compute the number of graphs consistent with the (observed + hypothetical) data after each experiment. We can then calculate the expected reduction in uncertainty after each stage in this experiment sequence. Note that we must compute the exact number of consistent graphs at each stage; simply finding a lower bound is insufficient.

Hence, determining the globally optimal sequence of experiments requires: (i) enumerating the factorial of an exponential number of sequences; (ii) for each sequence, constructing a tree of depth R with exponential branching structure; and (iii) for each branch-point and leaf in the tree, performing a probably exponential time calculation of the number of consistent graphs. This problem is clearly hopelessly intractable, probably even for very small numbers of genes.

Alternately, we might try to choose the best next experiment, even though this procedure may lead to a highly sub-optimal sequence of experiments. A straightforward algorithm is: For each novel experiment E, determine the $O_i = \sum_{s \in S} 2^{W(i,s)}$ possible data tables resulting from that experiment. For each possible outcome o, determine S_o, the cardinality of the set of graphs of interest, whether **InitialGraphs** or the set of consistent graphs. The score for an experiment is then just:

$$H_E = \frac{1}{|O|} \sum_{o \in O} S_o$$

Finally, the algorithm chooses the experiment with minimal H_E as the next experiment. This algorithm is greedy, and so will possibly pick out a sub-optimal (from a global perspective) series of experiments. It does, however, provide intuitively correct guidance for toy examples.

The formula can be weighted with prior probabilities on experiment outcomes, and weights on the experiment scores. Hence, it can accommodate prior beliefs about the likelihood that gene X regulates gene Y, as well as differential experiment cost (e.g., if experiments with more simultaneous interventions are more expensive than experiments with fewer).

For illustration, suppose we have only measured the wild type, and consider two experiments: E_1 = {knockout G_1}, and E_2 = {knockout G_1 & knockout G_2}. The possible outcomes of the experiments, and the corresponding sizes of the set of all graphs consistent with each outcome of each experiment are:

E_1:	G_1	G_2	G_3	**|Consistent Graphs|**
	–	0	0	64
	–	0	1	32
	–	1	0	32
	–	1	1	36

E_2:	G_1	G_2	G_3	**|Consistent Graphs|**
	–	–	0	64
	–	–	1	48

We find that H_{E1} = 41, and H_{E2} = 56; E_1 is selected as the better experiment

Throughout the above discussions, we have assumed that the variables not intervened upon in some novel experiment could possibly be significantly different (or not) from some other arbitrary experiment. This assumption does not always hold. For example, suppose we have three experiments E_1, E_2, and E_3. If we then consider some fourth experiment E_4 and a gene G not intervened upon in one of these four experiments, this assumption implies that there are no restrictions as to whether G is differentially expressed between E_4 and any of E_1 – E_3. However, suppose that the expression level of G in E_1, E_2, and E_3 is α_1, α_2, and α_3, respectively. If the α values are sufficiently far apart, then α_4 (the expression level in E_4) must be determined to be significantly different from at least one of the other α's (by whatever statistical test we are using). Hence, G must be differentially expressed between E_4 and at least one of E_1 – E_3, violating the assumption of our procedure for evaluating the informativeness of further experiments. Furthermore, we cannot compute in advance when the distribution of experimental values will "force" further differences, because certain data table completions can only be ruled out given information not present in the data table: namely, the extent of the differential expression between two experiments. The condition in the above example that the α values be "sufficiently far apart" is necessary, but not determinable from the input to the procedure for choosing the next experiment.

8 Statistical Tests for Real-World Data

Ideker *et al.* [2001] performed a series of experiments to investigate the galactose metabolism cycle in yeast (*Saccharomyces cerevisiae*). They focused on nine genes (*gal1*, *gal2*, *gal3*, *gal4*, *gal5*, *gal6*, *gal7*, *gal10*, and *gal80)* that prior work had identified as integral to the metabolization of galactose. They performed ten experiments (wild-type and single knockouts of each of the target genes) in two different environmental conditions (the presence and absence of galactose). We consider here only the experiments performed in the presence of galactose. The expression levels of approximately 5000 genes were then measured in each experiment, which consisted of four replications of both wild-type (control) and knock-out strain measurements, as illustrated in Table 1. We have reanalyzed the data from several statistical perspectives. The results of the analyses illustrate the complexity considerations described above, and they also indicate the fragility of statistical decisions with such small samples of so many variables.

Due to space constraints on the chips, the measured genes were divided into two subgroups, H1 and H2. There are thus two sets of chips: those that contain two spots for each gene in H1 (left side and right side), and those that contain two spots for each gene in H2 (left side and right side). The four measurements are distributed across four chips: two for the genes in H1, and two for the genes in H2. For a given gene in H1, for both spots on one of the two chips used for H1 genes, the controls are dyed red and the experimental condition is dyed green; for both spots on the other chip used to measure genes in H1, the controls are dyed green, and the experimental condition is dyed red. The same arrangement is used for the chips used to measure genes in H2.

For each gene and each experimental condition measurement, there is a control measurement of the same gene on the same spot on the same side of the same chip, using the opposite color – we will call the difference between two such measurements (or the difference of the logs of two such measurements) a *matched* difference. For each gene in H1, and each experiment, there are four matched differences; similarly for each gene in H2, and each experiment, there are four matched differences.

	Side	
Chip	Left	Right
1	H1: Green Exp. & Red Control *Difference 1*	H1: Green Exp. & Red Control *Difference 2*
2	H1: Red Exp. & Green Control *Difference 3*	H1: Red Exp. & Green Control *Difference 4*
3	H2: Green Exp. & Red Control *Difference 5*	H2: Green Exp. & Red Control *Difference 6*
4	H2: Red Exp. & Green Control *Difference 7*	H2: Red Exp. & Green Control *Difference 8*

Table 1: Chip Setup for Ideker, *et al.*

This setup produces some complex relationships among the measurement errors. In one of the experiments (henceforth referred to as the *null* experiment), both the control and the experimental condition are wild type in the presence of galactose (abbreviated as wt+gal). In this case, the average of the log differences between the green and the red measurements on the same spot should just be due to measurement noise. However, the measurements are correlated as shown in Table 2. Note that difference 1 and difference 2 are from the same chip, and are highly positively correlated; also difference 3 and difference 4 are from the same chip, and are also highly positively correlated. In contrast, difference 1 and difference 3 come from different chips, where the colors for the experimental conditions and the control conditions were reversed, and are negatively correlated.

	Diff. 1	Diff. 2	Diff. 3	Diff.4
Diff. 1	1.000	0.551	-0.499	-0.265
Diff. 2	0.551	1.000	-0.282	-0.553
Diff. 3	-0.499	-0.282	1.000	0.466
Diff. 4	-0.265	-0.553	0.466	1.000

Table 2: Correlations Among the Matched Differences in the Null Experiment

The FDR Procedure. Standard statistical tests control the probability of getting a false positive. If multiple tests are performed, unadjusted statistical tests produce incorrect results. Adjusted (e.g. by the Bonferroni adjustment which divides the p-value by the number of tests) procedures control the probability of getting *some* false positive out of any of the tests, but typically lead to tests of extremely low power. In contrast, the FDR (False Discovery Rate) procedure [Benjamini and Hochberg, 1995; Genovese and Wasserman, 2001] controls the percentage of positives that are false (in a distribution free manner) and produces tests that have much higher power. The FDR procedure takes as input a set of p-values, and a significance level. The significance level represents the maximum expected percentage of positives that are false positives. The FDR method chooses a threshold for the p-value that is between the Bonferroni threshold and the unadjusted threshold (alpha). All p-values less than this threshold are rejected. We have somewhat arbitrarily chosen a significance level of 0.05. The major problem is to produce a null distribution, which can generate the input p-values to the FDR procedure.

Application of FDR to the Data. For each of the nine genes involved in galactose metabolism, and for each pair of knock-out experiments, we performed a test to determine whether the hypothesis that the gene had the same expression level in the two experiments could be rejected. We here describe two different tests, both of which had the following features in common.

1. For each pair of experiments *A* and *B*, and each gene *X*, we compared the matched differences of gene *X* in experiment *A* (i.e. the expression level of gene *X* in the experimental condition on a given chip minus the expression level of gene *X* in the control condition on the same chip) to the matched difference of gene *X* in experiment *B*. Comparing the matched differences removes chip to chip variations in measured gene expressions.
2. Each test procedure that we ran produced 324 p-values (9 genes × (9 experiments choose 2)). The only difference between the procedures was in the tests that were run to produce the p-values, including differences in choice of null distributions for each hypothesis that there is no difference in the expression of a gene in two conditions.
3. The set of 324 p-values was given as input to the FDR procedure with a significance level of 0.05 to produce the final judgment about whether a particular gene was differentially expressed between two experiments.

Test #1: Constructed gene-pooled null distribution. In the Ideker *et al.* [2001] null experiment, both the experimental condition and the control condition on the same chip were from the wild type. We used this null experiment to derive the null distribution by the following procedure:

1. Put the matched differences for all the genes from each of the 4 replications of the null experiment into a single list.
2. Take 1000 samples of 8 matched differences from the list.
3. For each sample of 8, calculate the t-statistic for the first 4 of the 8 versus the second 4 of the 8. This gives a distribution of t-statistics for two samples of four differences from two distributions that are known to be the same.
4. Set the p-value for the test of the hypothesis that for a given gene the 4 matched differences from experiment *A* have the same mean as the 4 matched genes from experiment *B* to be the percentage of the 1000 t-statistics from the null that have absolute larger than the absolute value of the actual t-statistic.

The disadvantage of using these measurements for the null distribution is that all of the genes are pooled together to obtain the null distribution, which is therefore not specific to a given gene. The p-values generated in this way were passed into the FDR procedure, which found a number of significant differences. Substituting the Wilcoxon p-value for the t-statistic in the procedure described above produces the same results in 90% of the cases.

Test #2: Constructed gene-specific null distribution. There are a total of 40 measurements of the wt+gal control available for a null distribution for a given gene (ignoring the one experiment where wt+gal was both control and experiment), so it is possible to construct a null distribution for each gene using the following procedure for a given gene *X* and a fixed pair of experiments *A* and *B*.

1. Take all pairs of differences between the 40 different measurements of gene X in the control condition.
2. From the list of all pairs of differences, take a sample of 8 differences 1000 times.
3. For each sample of 8, calculate a t-statistic for the first 4 of the 8 versus the second 4 of the 8, and store the t-statistic. This provides a null distribution of 1000 t-statistics.

4. For gene X in experiment A and experiment B, calculate the actual t-statistic of the hypothesis that the 4 matched differences from experiment A have the same mean as the 4 matched statistic from experiment B.
5. Set the p-value equal to the percentage of the 1000 t-statistics whose absolute value is greater than or equal to the absolute value of the actual t-statistic.

The disadvantage of using the 40 measurements of the control for the null distribution is that a difference between two arbitrary controls varies different factors than the factors that are varied in each of the experiments. In particular, each of the elements in the null distribution comes from a difference between measurements on different chips, while the matched differences for experimental comparison come from within the same chip. These results agree with Test #1 89% of the time. The results of test procedures #1 and #2 are given in tables at:
http://www.phil.cmu.edu/projects/genegroup/tables/
danks2002_tables.pdf

9 Application of Algorithms to Yeast Data

The Ideker *et al*. [2001] data seem to be close-to-ideal for the algorithms considered in this paper, since we have expression data for all ten experiments (so we can perform all pairwise comparisons). Since all networks (cyclic or otherwise) are initially possible, there are $2^{72} \approx 4 \times 10^{21}$ possible genetic regulatory networks for these nine genes.[1] Clearly, any search over this space must either be automated or relatively arbitrary guesswork.

We applied the first two steps of the InitialGraphs algorithm to the data on expression differences between experiments derived from the Wilcoxon test using pooled genes (Test #1). The results are shown in table 3, where each row in each column gives the "*gal* numbers" for a minimal cover for the gene in that column.

This table describes 3,110,400 different graphs (some of which are cyclic). Using t-tests in Test #1 results in a comparable number of graphs. Step (3) of the IG procedure cannot be feasibly carried out (chiefly because of the cost of checking for redundant graphs), but running step 3 of the IG procedure on a sample of 2000 DAGs from step 2 yields an estimate of 1,185,062,400 total graphs from all three steps of the procedure.

Applied to the sparsest graph in **InitialGraphs**, the lower bound calculation described previously yields a lower bound on the order of 10^{18} graphs on 9 vertices consistent with the experimental data. These ten experiments have thus reduced the hypothesis space by (at most) three orders of magnitude.

gal1	*gal2*	*gal3*	*gal4*	*gal5*	*gal6*	*gal7*	*gal10*	*gal80*
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 3: Output of the IG Procedure

10 Previous Work

At least two algorithms for learning gene regulation networks from gene perturbation experiments [Onami *et al.* 2001; Ideker *et al.*, 2000] have been previously proposed, and some of the algorithms described above were inspired by the Ideker *et al.* paper. It is important to consider them because they suggest very optimistic conclusions about gene perturbation experiments as a search strategy. Both proposals assume the data are projected to binary values, an assumption unfaithful to the statistics. Specifically, one cannot just compare the wild-type expression of gene *Y* with its expression level in experiments in which *X* is knocked out or overexpressed. Suppose three experiments yield the following data table, where wt_i denotes the level of expression of the wild type for gene *i*, and ε, δ are positive quantities.

Gene *X*	Gene *Y*
wt_X	wt_Y
knocked out	$wt_Y + \varepsilon$
overexpressed	$wt_Y - \delta$

The differences ε and δ may not be statistically significant, but $\varepsilon + \delta$ may very well be statistically significant. Suitable variations in the expression of gene *X* may produce variations in the expression of gene *Y*, but comparisons with the wild-type alone may not reveal it. Regulatory networks seem not to be Boolean networks. Binarizing the data can thus obscure significant expression differences. Whatever threshold is chosen to divide the expression of a gene between high and low, there can be significant differences between values on the same side of the threshold, and insignificant differences between variables on opposite sides.

The Difference-Based Regulation Finding (DBRF) method of Onami *et al.* [2001] is as follows: Start with measurements of *n* genes in *k* experiments, each of which has one gene (or none – i.e., wild type) enhanced or suppressed. Then form a directed graph with one node per

[1] Since there can (independently) be an edge or not between any of the $n*(n–1)$ ordered pairs of genes.

gene, and a directed edge from node *X* to node *Y* if and only if the expression level of gene *Y* is different from wild-type when gene *X* is knocked out or overexpressed. Mark each directed edge with a '+' if overexpression produces overexpression or if suppression produces suppression, and '–' otherwise.

This graph will include edges corresponding to both direct and indirect regulation. To remove the edges due to indirect regulation, consider the set of all (presumably acyclic) paths from *X* to *Y* for all *X*, *Y*. Define the parity of each path be the ordinary sign of the product of signs over all of the directed edges in the path. For each parity, eliminate all of the paths from *X* to *Y* except the longest one(s).

Because the DBRF algorithm keeps only the longest paths, it can only find the correct structure if all upregulatory and downregulatory pathways from gene *X* to gene *Y* are of the same length (i.e., involve the same number of intermediate genes). Furthermore, the restriction to single knockouts amounts to an assumption that no gene's regulation is masked by the influence of another gene. For example, there cannot be disjunctive dependencies in which gene *Z* is expressed if either gene *X* or *Y* is. The authors suggest that their algorithm can be extended to multiple knockouts, but it is unclear how this can be done without introducing further errors. Thus, the DBRF method is provably incorrect: there are (biologically plausible) regulatory networks it cannot learn.

Ideker, Thorsson, and Karp [2000] (henceforth, ITK) assume the true regulatory structure determines an acyclic Boolean network, in which the values for each node are "high" and "low" expression, coded as 1 and 0 respectively, and the expression level of each node of positive indegree is a Boolean function of its parents. Arrange the binarized values in an (experiment × [gene & exogenous factor]) matrix ***E***, replacing the 0/1 with a – or + for any matrix entry (*j*, *k*) in which gene *k* has been exogenously suppressed or enhanced in experiment *j*.

For each gene *Y*, consider all pairs of rows in ***E*** in which there is no manipulation of *Y* in either row, and *Y*'s expression differs between the two rows. For each pair (*A*, *B*) of rows, let C_{AB} be the set of genes whose expression level also differs between rows *A* and *B*. Define C_{min} to be the smallest covering set of all of the C_{AB}: (i) for each C_{AB}, at least one element of C_{min} is in C_{AB}; and (ii) if we remove any element from C_{min}, property (i) no longer holds. Note that there may be multiple C_{min}. The set of possible network structures then contains all possible graphs formed by choosing one C_{min} for each gene *Y* and including edges from the variables in C_{min} to *Y*. The Boolean function for a gene *Y* in a particular network is then determined by filling in the truth table for *Y* using the values in ***E***. Values not determined by ***E*** are encoded with an '*'.

To illustrate, suppose the regulatory network dependencies are as follows, with the dependent variables on the left:

B := *A*; and

D := 1 if and only if *A* = *B* = *C*.

That is, the true regulatory graph is:

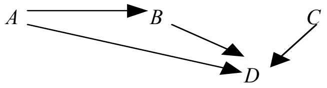

Figure 1

And the data table ***E*** is:

	A	*B*	*C*	*D*
wt	1	1	1	1
E1	-	0	1	0
E2	1	-	1	0
E3	1	1	-	0
E4	1	1	1	-

Except when exogenously manipulated, *A* never changes, so its minimal set is empty. The same is true of *C*. Every time *B* changes non-exogenously, *A* changes as well and there is a pair of experiments for which only *B* and *A* change. Therefore, the minimal set for *B* is {*A*}. When *D* changes, then either *B* changes or else *C* changes, and there is a pair of experiments for which only *B*, *D* change and another pair for which only *C*, *D* change. Hence, the minimal covering set for *D* is {*B*, *C*}. Since each gene has exactly one minimal set of parents, there is a unique graph, which is:

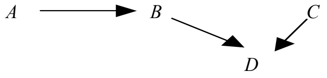

Figure 2

And the truth tables determining B and D, so far as they can be estimated from the data, are:

A	B		B	C	D
1	1		1	1	1
0	0		0	1	0
			1	0	0
			0	0	*

ITK also provide an algorithm for choosing the best next experiment, given some set of *N* possible regulatory networks. For each possible experiment *p*, we first determine the outcome of the experiment predicted by each given network. If a variable value is not predicted by a network (because of an '*' in a truth table), then we randomly choose a value for that variable. This procedure will generate *S* (≤ *N*) distinct predictions *s*, and define N_s to be the number of networks that make prediction *s*. We then choose the experiment *p* that maximizes the entropy score:

$$H_p = -\sum_{s=1}^{S} \frac{N_s}{N} \log_2\left(\frac{N_s}{N}\right)$$

The experiment chooser is often unhelpful (or even misleading). Applied to data table **E**, if the value of a random choice of the single '*' is 0, we obtain $D := B \& C$. If the value of a random choice of '*' is 1, we obtain $D := 1$ if and only if $B = C$. In either case, every possible experiment determines a unique resulting state, and so the entropy scores for all experiments are the same. In general, the experiment chooser offers no guidance whenever only one network is provided. Even when multiple graphs are provided, they may not include the true graph, and so the sequence of experiments determined by the entropy formula above may constitute a very sub-optimal route to the truth.

ITK restrict their procedure to acyclic graphs, but of course feedback systems are ubiquitous in gene regulation. There are special problems for a minimal covering set procedure (such as ITK) if cyclic networks are allowed. For example, suppose the true structure is

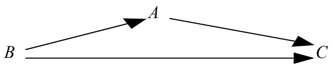

Figure 3

where $C := A \& B$ and $A := B$, and the (binarized) data **E'** are:

	A	*B*	*C*
wt	1	1	1
E1	-	1	0
E2	0	-	0

The minimal cover for B is the empty set, the minimal cover for C is $\{A\}$, but A has two minimal covers, namely $\{B\}$ and $\{C\}$. Thus the minimal cover procedure alone produces two graphs:

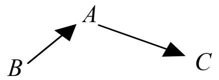

Figure 4

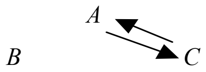

Figure 5

But inferences from the data table *ought* to eliminate the second of these graphs, since manipulation of B alters both A and C, which is not predicted in Figure 5. Thus, if cyclic graphs are allowed, the minimal cover set procedure needs to be supplemented with the principle that if an intervention (as in our clause (iii) for the IG algorithm) on a set **S** of variables changes a variable Y, then there must be a directed path from some member of **S** to Y. With this supplementation, the data table is consistent with Figure 4, as well as the graphs:

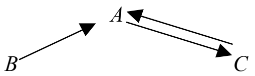

Figure 6

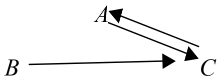

Figure 7

Note that neither of these two graphs is minimally consistent with the data.

11 Conclusion

The vague but important difference between search methods and confirmatory methods is this: search methods start with a very large set of possible hypotheses, and attempt to locate a much smaller set of hypotheses containing the truth (or whatever is an acceptable approximation to the truth), while confirmatory methods start with a very specific hypothesis and attempt to establish or refute it. The vast hypothesis space of possible regulatory networks requires trustworthy, feasible search methods. Covariational methods are feasible, but not trustworthy with present measurement technologies. The results in this paper argue that experimental differences from gene perturbations do not constitute a feasible search method.

At the same time, search for genetic regulatory networks often (in practice) brings together a wide range of information and models. These results demonstrate the ineffectiveness of gene perturbation experiments by themselves, but they may prove useful in combination with other sorts of data. For example, immuopreciptiation/gene location methods [Lee *et al.*, 2002] now offer the ability to identify genes regulated directly by known regulators, and the next major improvement in the combinatorics of gene regulation may well adapt these techniques to the simultaneous measurement of direct effects of all regulators. Gene perturbation experimentation can be expected to remain an important strategy for confirming the results of these and other search strategies.

Acknowledgements

Research for this paper was supported by NASA Ames Research Center NRA A2-37143 and agreement NCC 2-1295. We thank Larry Wasserman for very helpful conversations concerning the statistical testing procedures described in this paper, and the Institute for Systems Biology, Seattle, Washington, for providing the data reanalyzed here.

References

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

[Chu, 2003] Tianjao Chu. *Learning from SAGE Data*. Ph.D. dissertation, Dept. of Philosophy, Carnegie Mellon Univ.

[Genovese and Wasserman, 2001] Christopher Genovese and Larry Wasserman. False Discovery Rates, Technical Report 762, Dept. of Statistics, Carnegie Mellon Univ.

[Ideker *et al.*, 2000] Trey E. Ideker, Vesteinn Thorsson, and Richard M. Karp. Discovery of Regulatory Interactions through Perturbation: Inference and Experimental Design. *Pacific Symposium on Biocomputing*.

[Ideker *et al*., 2001] T. E. Ideker, V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. R. Goodlett, R. Aebersold, and L. Hood. Integrated Genomic and Proteomic Analyses of a Systematically Perturbed Metabolic Network. *Science* 292:929-934.

[Lee *et al*., 2002] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. R. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional Regulatory Networks in *Saccharomyces cerevisiae*. *Science* 298:799-804.

[Onami *et al*., 2001] S. Onami, K. M. Kyoda, M. Morohashi, and H. Kitano. The DBRF Method for Inferring a Gene Network from Large-Scale Steady-State Gene Expression Data. In H. Kitano (ed.), *Foundations of Systems Biology*. Cambridge, Mass.: The MIT Press, pp. 59-75.