

# An Evaluation of a System that Recommends Microarray Experiments to Perform to Discover Gene-Regulation Pathways

Changwon Yoo      [cwyoo@vbi.vt.edu](mailto:cwyoo@vbi.vt.edu) / Tel: 540-231-2100  
Virginia Bioinformatics Institute, Virginia Polytechnic and State University  
1880 Pratt Drive, Building XV, Blacksburg, VA 24061

Gregory F. Cooper      [gfc@cbmi.upmc.edu](mailto:gfc@cbmi.upmc.edu)  
Center for Biomedical Informatics, University of Pittsburgh  
8084 Forbes Tower, 200 Lothrop St., Pittsburgh PA 15213

## Abstract

The main topic of this paper is modeling the expected value of experimentation for discovering causal pathways in gene expression data. By experimentation we mean both interventions (e.g., a gene knock-out experiment) and observations (e.g., passively observing the expression level of a “wild-type” gene). We introduce a system called GEEVE (causal discovery in Gene Expression data using Expected Value of Experimentation), which implements expected value of experimentation in discovering causal pathways using gene expression data. GEEVE provides the following assistance, which is intended to help biologists in their quest to discover gene-regulation pathways:

- Recommending which experiments to perform (with a focus on “knock-out” experiments) using an expected value of experimentation (EVE) method.
- Recommending the number of measurements (observational and experimental) to include in the experimental design, again using an EVE method.
- Providing a Bayesian analysis that combines prior knowledge with the results of recent microarray experimental results to derive posterior probabilities of gene regulation relationships.

In recommending which experiments to perform (and how many times to repeat them) the EVE approach considers the biologist's preferences for which genes to focus the discovery process. Also, since exact EVE calculations are exponential in time, GEEVE incorporates approximation methods. GEEVE is able to combine data from knock-out experiments with data from wild-type experiments to suggest additional experiments to perform and then to analyze the results of those microarray experimental results. It models the possibility that unmeasured (latent) variables may be responsible for some of the statistical associations among the expression levels of the genes under study.

To evaluate the GEEVE system, we used a gene expression simulator to generate data from specified models of gene regulation. The results show that the GEEVE system gives better results than two recently published approaches (1) in learning the generating models of gene regulation and (2) in recommending experiments to perform.

Keywords: Causal discovery; Systems biology; Causal Bayesian networks; Microarray study design

## **1 Introduction**

Most research on causal discovery using causal networks has been based on using passive observational data [6, 17, 42]. There are limitations in learning causal relationships from observational data only. For example, if the generating process contains a latent factor (confounder) that influences two variables, it can be difficult, if not impossible, to learn the causal relationships between those two variables from observational data alone.

To uncover such causal relationships, a scientist generally needs to design a study that involves manipulating a variable (or variables) and then observing the changes (if any) in other variables of interest. In such an experimental study, one or more variables are manipulated and the effects on other variables are measured. On the other hand, *observational data* result from passive (i.e., non-interventional) measurement of some system, such as a cell. In general, both observational and experimental data may exist on a set of variables of interest. Limited time and funds restrict the number of variables that can be manipulated and the number of *experimental repeats* that can be collected for the control and experimental groups. For example, a molecular biologist who is interested in discovering the causal pathway of the genes involved in galactose metabolism first has to select the genes he or she is interested in understanding at a causal level. These genes are usually selected based on previously published results or by the molecular biologist's scientific interest. Many issues are considered in determining the number of experimental repeats to obtain for each variable in the study design. Having more experimental repeats will typically tighten the statistical confidence intervals in the data analysis. Considering available time, budget, and other constraints, the biologist will make a decision about the number of experimental repeats to obtain.

Developing causal analysis methods is a key focus of several fields. In statistics, jointly with medicine, issues related to randomized clinical trials (RCTs) are studied, including methods for finding an optimal number of cases using stopping rules [3, 12, 41]. In molecular biology, developing techniques that generate efficient experimental designs for high throughput methods, such as cDNA microarrays, is gaining interest [26, 30]. In artificial intelligence, techniques using

graphical models have been used to model experimentation and have been applied to suggest the next experiment for causal discovery [23, 27, 45].

All these prior approaches have made contributions to efficient causal study design (see Section 2 for details). They are not, however, sensitive to issues of limited resources and experimenter preferences. The research reported here is concerned with developing and evaluating a decision-analytic system that considers these issues in helping a biologist design and analyze studies of cellular pathways using high throughput sources of data. In particular, this paper concentrates on the design and analysis of cDNA microarray studies for uncovering gene regulation pathways. The fundamental methodology, however, is applicable to analyzing other high throughput data sources, such as the measurement of protein-levels, which is a rapidly developing area of biology.

The GEEVE (causal discovery in Gene Expression data using Expected Value of Experimentation) system uses ideas from different areas of study. GEEVE uses causal Bayesian networks (see Section 2.1) and incorporates an experimenter's preference (see Section 2.2) to give recommendations to the experimenter about designing a gene expression experimental study (see Section 2.3). In the remainder of this section, we provide background on gene array chips and give an overview of the problem addressed by the GEEVE system.

## **1.1 Gene array chips**

Three major gene-expression measurement technologies are currently available for measuring the expression levels of many genes at once. One is called a cDNA microarray, or simply *DNA*

*microarray* [4]; another is called an *oligonucleotide array*, or GeneChip® [31]; and a third technique is called *SAGE* (Serial Analysis of Gene Expression). We concentrate in this paper on the first two techniques, since they are high throughput methods, whereas SAGE is a more time consuming method. The DNA microarray technique uses user-definable probes<sup>1</sup> on the microarray, and the oligonucleotide array uses small oligonucleotide (usually 200 or 300 bases) as factory-built probes.

## 1.2 Problem description

A gene expression study using DNA microarrays usually involves two major steps. The first step typically consists of performing initial experiments to narrow the set of genes to study in more detail. The experimenter can avoid this first step if he or she already knows the specific set of genes of interest. Since the functions of many genes are not known, the first step is usually necessary. A number of microarrays will be assigned to hybridize with a pool of controlled cells and experimental cells. By examining the genes that are differentially expressed in these two groups of cells, the experimenter can decide which genes to study further. After choosing a set of genes, the experimenter needs to produce an experimental design to investigate how those genes are functionally related to each other.

## 2 GEEVE System

This section describes the issues related to the implementation of the GEEVE system. Tong and Koller [45] used a single-case approach to recommend to the experimenter the best possible pairwise relationship for further investigation. In gene expression microarray studies, it may not

---

<sup>1</sup> According to the nomenclature recommended by B. Phimister of *Nature Genetics*, a *probe* is the nucleic acid with known sequence, whereas a *target* is the free nucleic acid sample whose abundance level is being detected.

be practical to perform one experiment at a time. Often it is more efficient to repeat a given experiment multiple times in parallel, rather than to repeat the experiment sequentially over time.

Tong and Koller [45] and Ideker et al. [23] used edge entropy loss functions to search for the next best experiment to perform. This approach focuses strictly on information gain as a utility measure; using it can be useful when the experimenter is performing a first-phase study to select the genes without any preference toward the relationships among the genes. After the first-phase study, however, the experimenter will usually have some preference for which genes to study in greater detail. As more gene expression experiments (studies) are performed, the experimenter will refine his or her preferences about the relationships to study in more and more detail.

Consequently, a recommendation system that incorporates the preferences of the experimenter seems desirable.

GEEVE allows for repeats of an experiment, and it can be sensitive to an experimenter's preferences for which genes to study. These improvements ostensibly make GEEVE more applicable to real-world design of gene expression experiments. GEEVE also incorporates an efficient causal discovery method that is based on an extension of a causal discovery algorithm [50].

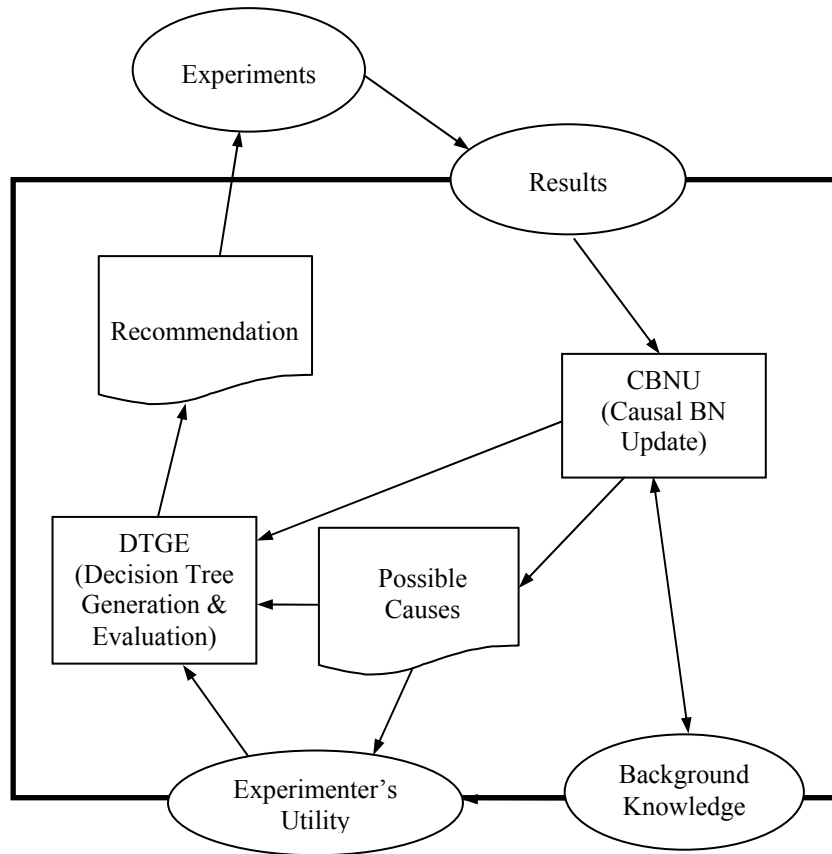


Figure 1. The GEEVE system. The box with the thick line represents the GEEVE system. Boxes in GEEVE represent system modules. Boxes with wavy lines on the bottom represent outputs from GEEVE. The oval labeled Experiments is an object that is outside of GEEVE. The ovals on the GEEVE border represent objects that communicate with GEEVE from the outside.

The GEEVE system consists of two modules called the causal Bayesian network update (CBNU) module and the decision tree generation and evaluation (DTGE) module (Figure 1). The CBNU module uses an algorithm called *Implicit Latent Variable Scoring* (ILVS) method [50] to causally analyze the current microarray data in light of the user’s prior beliefs about causal relationships among the genes under study. The DTGE module evaluates a decision tree that was generated based on the results of the CBNU module and the experimenter’s preferences, which are expressed with GEEVE as a utility function. Finally (under assumptions) the best possible experiments are recommended to the experimenter. The experimenter then chooses the next

experiment to perform, which may or may not be the one suggested by GEEVE. When the results are available, they can be submitted to the CBNU module for a new round of analysis.

## 2.1 Updating causal Bayesian networks

This section describes a new method to evaluate causal Bayesian networks using a mixture of observational and experimental data. The algorithm described in the current section is incorporated into the GEEVE system.

A causal Bayesian network (or *causal network* for short) is a Bayesian network in which each arc is interpreted as a direct causal influence between a parent node (variable) and a child node, relative to the other nodes in the network [34]. Figure 2 illustrates a hypothetical causal Bayesian network structure containing five nodes that represent genes. The probabilities associated with this causal network structure are not shown.

The causal network structure in Figure 2 indicates, for example, that the *Gene1* can regulate (causally influence) the expression level of the *Gene3*, which in turn can regulate the expression level of the *Gene5*. The causal Markov condition gives the conditional independence relationships specified by a causal Bayesian network:

*A variable is independent of its non-descendants (i.e., non-effects) given its parents (i.e., its direct causes).*

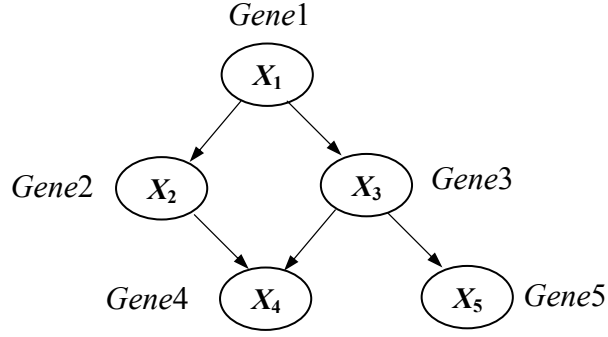


Figure 2. The structure of a causal Bayesian network that represents a portion of a hypothetical gene-regulation pathway.

The causal Markov condition permits the joint distribution of the  $n$  variables in a causal Bayesian network to be factored as follows [34]:

$$P(x_1, x_2, \dots, x_n | K) = \prod_{i=1}^n P(x_i | \pi_i, K) \quad (1)$$

where  $x_i$  denotes a state of variable  $X_i$ ,  $\pi_i$  denotes a joint state of the parents of  $X_i$ , and  $K$  denotes background knowledge.

We introduce six equivalence classes ( $E_1$  through  $E_6$ ) among the structures (Figure 3). The causal networks in an equivalence class are statistically indistinguishable for any observational and experimental data on  $X$  and  $Y$  where  $H$  represents a latent variable, which might represent (for example) the mRNA expression level of an unmeasured gene.

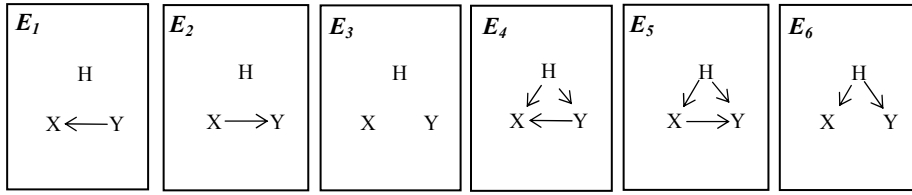


Figure 3. Six Local Causal Hypotheses

Using the previously published structure scoring method [6, 18], we introduced the ILVS method to score the six hypotheses in Figure 3 [50, 51]. *Local ILVS Method (LIM)* was introduced to score structures with more than pairwise variables [50]. A high level pseudo code is given in Figure 4. To analyze the microarray datasets LIM uses (1) the dataset of gene expression level under different experimental conditions as an input; and (2) the structure scores of the local structures that consists of  $k$  genes ( $k > 2$ ) as an output. For example, assume there are four genes that are being modeled, i.e.,  $V = \{G_1, G_2, G_3, G_4\}$  and  $k = 3$ , then for all 12 gene pairs,  $R_i = \{G_j, G_l\}$ ,  $j, l \in \{1, 2, 3, 4\}, j \neq l$ , LIM (1) searches for additional variables to include in the local structure (since  $k = 3$ , we only have to search for one additional variable) by selecting the most correlated variables with  $G_j$  and  $G_l$ . (2) performs an anytime structure search on a local structure with  $k$  variables. (3) scores the six hypotheses  $E_i$  (see Figure 3) based on the local structured visited and scored. Thus, LIM is capable of analyzing datasets with large number of variables, e.g., microarray datasets (>5,000 genes). More detail information of ILVS and LIM can be found at Yoo and Cooper [48] and Yoo [49].

```

For each  $(X, Y)$ ,  $X \neq Y$  and  $X, Y \in \{\text{All modeled variables}\}$ 
   $\Omega \leftarrow$  Select the most correlated variables of  $(X, Y)$  subject to  $|\Omega| < k$ ;
  For  $i = 1$  to 6
     $S \leftarrow$  Greedy hill climbing structure search with variables in  $\Omega$  and  $X$  and  $Y$ ;
    Perturb  $S$  and perform Greedy Hill Climbing Structure Search on  $S$  within the user-defined number of iterations;
    Score  $E_i$  using ILVS and model averaging;
  EndFor
  Normalize score of  $E_i$  for  $(X, Y)$ ;
EndFor

```

Figure 4. A high level pseudo code of LIM. Note that  $S$  is a *local structure* in which it does not include all modeled variables (set  $\Omega$  is limited to include only  $k$  variables).

## 2.2 GEEVE Utility Model

GEEVE is capable of incorporating an experimenter’s utility model [49]. In the research reported in this paper, we do not explore this aspect of GEEVE, because we empirically compare GEEVE’s performance to other methods that do not allow modeling utilities flexibly. Instead, we used the following utility assumptions, where  $E_i^{XY}$  denotes the node pair  $X$  and  $Y$  with causal relationship  $E_i$ : (1) For all pairs  $(X, Y)$ ,  $U(X, Y) = 0.5$ , which means that all gene pairs are of equal interest; (2)  $U(E_i^{XY} | E_j^{XY}) = 1$  for all  $i = j$ , which means that when the predicted structure  $E_i^{XY}$  matches the generating structure  $E_j^{XY}$ , the utility is assigned to be the highest possible value (=1.0); (3)  $U(E_i^{XY} | E_j^{XY}) = 0.5$  for all  $E_i^{XY}$  and  $E_j^{XY}$  that have equivalent causal relationships, not including consideration of a latent confounder ( $E_1^{XY}$  and  $E_4^{XY}$ ,  $E_2^{XY}$  and  $E_5^{XY}$ , and  $E_3^{XY}$  and  $E_6^{XY}$  have equivalent causal relationships plus or minus a latent confounder); and otherwise (4)  $U(E_i^{XY} | E_j^{XY}) = 0$ .

In general, the GEEVE utility for reporting the relationship  $E_i^{XY}$  to the user (experimenter) is derived as follows. The weights  $w_{ij} = U(E_i^{XY} | E_j^{XY})$  are used as a shorthand notation. The

following term is then derived:  $q_i = \sum_j w_{ij} \cdot P(E_j^{XY} | D, K)$ , where the probability term is output by

LIM. Finally, the experimenter's utility for discovering a novel and interesting causal relationship is calculated as  $q_i \cdot U(X, Y)$ .

### 2.3 Generating a decision tree

Based on the experimenter's utility specification and the causal Bayesian network output by LIM, the GEEVE system builds a decision tree and evaluates it. GEEVE concentrates on pairwise relationships of genes and generates the following decision tree:

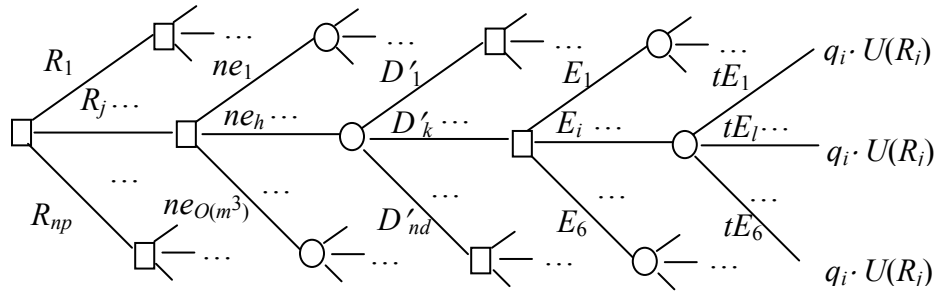


Figure 5. A GEEVE decision tree.

where (1)  $R_j$  represents an arbitrary pair of genes and  $np$  represents the number of pairs of the genes being modeled, (2)  $ne_h$  represents the experimental conditions (explained below) and  $m$  represents a maximum number of gene-expression measurements that are obtained for an experimental study, [ $O(m^3)$  also explained below], (3)  $D'_k$  represents the outcome of investigating a given pair of genes, (4)  $E_i$  denotes a decision to claim structure  $E_i$  (see Figure 3) as representing the causal relationship between a given pair of genes, (5)  $tE_i$  represents that the correct structure is  $E_i$ , and (6)  $q_i$  is as defined in Section 2.2.

For the decision tree shown in Figure 5, assume that there are (1) at most  $s$  states for each gene-expression-level variable, (2)  $v$  variables that model gene expression levels, and (3)  $m$  microarray experiments (wherein each variable has some state). Then the number of possible datasets  $nd$  is  $O(s^{vm})$ . LIM uses a simulation method [21] to make the number of possible datasets manageable. It keeps track of the highest scoring local structure relative to a given set of experimental conditions and previous data  $D$ . Using the highest scoring local structure, GEEVE generates possible future experimental results, which are shown in Figure 5 as  $D'_1, D'_2, \dots, D'_{nd}$ .

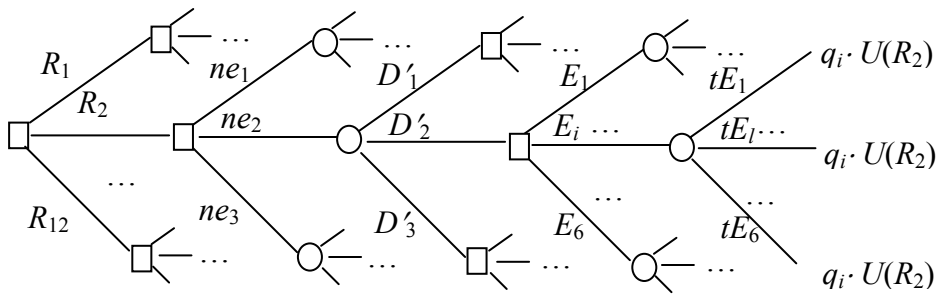


Figure 6. An example of a decision tree

We use the same example used in Section 2.1, i.e., the gene pairs  $R_i = \{G_j, G_l\}$ . Since we are analyzing four genes, there are 12 unique pairs, i.e.,  $R_1 = \{G_1, G_2\}, R_2 = \{G_1, G_3\}, \dots, R_{12} = \{G_3, G_4\}$ . For simplicity, assume there are three possible experiments for each pair of genes. For example, following the  $R_2$  branch in Figure 6, the three possible experiments are (1) one wild type measurement of  $G_1$  and  $G_3$  ( $ne_1$ ); (2) one knockout of  $G_1$  and measure  $G_3$  ( $ne_2$ ); and (3) one knockout of  $G_3$  and measure  $G_1$  ( $ne_3$ ). Further assume that LIM searched the following structure as the local structure of  $R_2 = \{G_1, G_3\}$  that best fits the data  $D$  that were collected so far:



LIM parameterizes the above local structure based on  $D$  and uses simulation method [21] to generate simulated data. For example, the simulated dataset (e.g.,  $D'$ ) after the experiment  $ne_2$  (one knockout of  $G_1$  and measure  $G_3$ ) in Figure 6 might be simulated as the following:

Genes Measured cases	$G_1$	$G_3$	$G_4$	Note
1	1	2	1	Previous Data
2	0	1	2	
...				
$ D $	1	2	2	
added	0	1	1	Simulated Data

where each row of the table represents a measurement from an experiment. We labeled each measurement with indices  $1, 2, \dots, |D|$ . The middle three columns represent the expression level of each gene (0 = low expression, 1 = no change, 2 = high expression). Note that since the experiment  $ne_2$  measures only one case, one simulated case is added in the dataset  $D$  (shaded row in the above table). In Figure 6, we are assuming that the user only wants to generate three datasets for the simulation ( $nd = 3$ ). For each simulated dataset, we again use LIM to score the six hypotheses  $E_i$  (see Figure 3) and further calculate  $q_i$  and complete the decision tree.

Solving the decision tree in Figure 5 is exponential in the number of microarray experiments (cases). Therefore, we need an approximation method to evaluate the decision tree. Several

different approximation methods are available with some assumptions [5, 16]. Heckerman et al. [16] introduced a non-myopic approximation method assuming that for a large decision tree, the central limit theorem holds. The method was non-myopic in the number of chance nodes but not in the number of decision nodes. Chavez and Henrion [5] assumed additive expected utility independence and used linear regression to estimate the expected value of perfect information (EVPI) and expected value of information (EVI). However, Heckerman et al. [16] and Chavez and Henrion [5] approximations are not suitable with large number of decision branches because they assume binary decision nodes. Thus, we use a random heuristic search to approximate the expected value of experimentation, as explained next.

GEEVE models possible experimental conditions for node pair  $X$  and  $Y$  as (1) passively observing  $X$  and  $Y$ , e.g., a wild type experiment; (2) manipulation of  $X$ , e.g., a knock out experiment, or (3) manipulation of  $Y$ . If the maximum number of allowed experiments in an overall experimental protocol is  $m$ , the number of possible protocols (which we also call experimental conditions) is  $O(m^3)$ .

To enable an efficient search, GEEVE defines operations on the number and type of microarray experiments in a protocol that is intended to discover the causal relationships between genes  $X$  and  $Y$ . Let set  $ne_h = (m_O, m_X, m_Y)$  represent the experimental condition where the first element  $m_O$  is the number of measurements in which both  $X$  and  $Y$  are both passively observed (e.g., a “wild type” measurement); the second element  $m_X$  is the number of measurements in which  $X$  is manipulated and  $Y$  is observed; and the third element  $m_Y$  is the number of measurements in

which  $Y$  is manipulated and  $X$  is observed. The operators used in GEEVE's heuristic greedy hill climbing search for the best setting of the parameter vector  $ne_h = (m_O, m_X, m_Y)$  are as follows:

- $MZ(ne_h, i)$  : set the  $i$ -th element in  $ne_h$  to zero
- $DH(ne_h, i)$  : decrease the  $i$ -th element in  $ne_h$  by half
- $DD(ne_h, i)$  : double the  $i$ -th element in  $ne_h$

Using these operators, GEEVE performs the following heuristic search for the value of the parameters  $(m_O, m_X, m_Y)$ :

- Step 1: The initial parameter values that are tried in the decision tree as follows:

$$\left\{ m - \left\lfloor \frac{2m}{3} \right\rfloor, \left\lfloor \frac{m}{3} \right\rfloor, \left\lfloor \frac{m}{3} \right\rfloor \right\}, \left\{ 0, m - \left\lfloor \frac{m}{2} \right\rfloor, \left\lfloor \frac{m}{2} \right\rfloor \right\}, \left\{ m - \left\lfloor \frac{m}{2} \right\rfloor, 0, \left\lfloor \frac{m}{2} \right\rfloor \right\}, \left\{ m - \left\lfloor \frac{m}{2} \right\rfloor, \left\lfloor \frac{m}{2} \right\rfloor, 0 \right\},$$

$\{m, 0, 0\}$ ,  $\{0, m, 0\}$ , and  $\{0, 0, m\}$ . Choose the experimental condition  $ne_h^* = \{m_O^*, m_X^*, m_Y^*\}$

that has highest expected value.

- Step 2: Set  $ne_h$  to be  $ne_h^*$ .
- Step 3: Apply  $MZ(ne_h, i)$ ,  $DH(ne_h, i)$ , and  $DD(ne_h, i)$  for  $i=1, 2, 3$ .
- Step 4: Evaluate expected value for all experimental conditions in Step 2 and Step 3 and choose the experimental condition  $ne_h' = \{m_O', m_X', m_Y'\}$  that has highest expected value. If the expected value of  $ne_h'$  is higher than  $ne_h^*$  then let  $ne_h^*$  be  $ne_h'$  and repeat Step 2; otherwise randomly select  $ne_h = \{m_O, m_X, m_Y\}$  where  $m_O + m_X + m_Y = m$  and go to Step 3 if the repetition is smaller than some user-defined threshold.

The best experiment found by GEEVE when it completes its heuristic evaluation of the decision tree will be the experimental condition  $ne_{\max} = \arg \max_{h \in \{1, 2, \dots, O(m^3)\}} ne_h^*$  on the gene pair  $R_j$ , where  $h$  is the index of  $ne_h^*$  that yielded  $ne^*$ .

### 3 Related Work

The GEEVE system incorporates an experimenter's preferences into a decision model in order to give recommendations about designing a gene-expression experimental study. The decision model it uses is based on decision theory [29, 47]. Many different fields concentrate on study design for causal discovery. Traditionally, in statistics and medicine, research on causal discovery is actively pursued in research on controlled trials [1, 3, 41]. In computer science, causal discovery is also an active research topic, especially in the machine learning community [6, 17, 19, 35, 42, 49]. In biology, recent microarray technologies have fueled a field known as *systems biology*, which seeks to discover causal relationships among a large number of genes and other cellular constituents [24, 40]. In this section, we will briefly review work related to this paper, concentrating especially on the fields just mentioned.

#### 3.1.1 Genetic pathway models

Before describing pathway models, we first place them in the context of gene clustering methods, which have been very popular the last few years. Indeed most of the early work on gene expression data analyses used clustering methods. Gene expression levels that were measured by cDNA microarray in the yeast cell-division cycle were analyzed for the first time using a cluster analysis [40]. A cluster analysis typically searches for groups of genes that show a similar expression pattern under different experimental conditions. Other analyses followed

using similar cluster analyses applied to microarray data [15, 22, 32]. Cluster and classification analyses do not necessarily provide causal information, which is at the heart of gene pathway discovery. In contrast, knowledge of causal pathways can be used to produce a causal clustering of the genes.

Tsang [46] and Dutilh [9] each give a review of genetic networks. A review that is focused more on modeling methods is given by de Jong [8]. A thorough review based on biological context was published by Smolen et al. [39], who suggested that current microarray techniques are limited in delineating intracellular signaling pathways. Smolen et al. [39] argues that since microarray technology is measuring an average expression level of a gene among millions of cells, there is little we can learn about gene-regulation pathways information from the data. We will discuss this issue in Section 5.2 with respect to latent variable detection.

### **3.1.2 Experimentation recommendation models**

Computational models of scientific discovery were actively studied in artificial intelligence (in conjunction with psychology) in the late 1980s [38]. In molecular biology in particular, Karp [25] created systems in bacterial gene regulation that could describe the initial conditions of an experiment, generate a hypothesis, and refine it. We will describe additional systems in Sections 3.1.2.1 and 3.1.2.2 in more detail because they will be used as points of comparison when evaluating GEEVE in Section 4.

#### **3.1.2.1 Active learning in Bayesian networks**

An extension of supervised learning, *active learning* was applied to learning causal Bayesian networks in scientific discovery [45]. Tong and Koller used edge entropy loss functions and a

myopic search in order to recommend the next best experiment to perform. Their main assumptions are: (1) discrete variables only; (2) no missing data; and (2) no modeling of latent (hidden) variables. They modeled experimental manipulation using the manipulation representation in Cooper and Yoo [7].

Tong and Koller applied their algorithm to three Bayesian networks with 5, 8, and 16 nodes respectively. Based on their simulations, they showed that active learning performs better in finding BN structures than randomly choosing of the query nodes.

### 3.1.2.2 Entropy score and set covering in Boolean networks

Ideker et al. (2000) used binary networks to model the perturbation on a gene network and used an entropy loss function to recommend the next best perturbation to perform, where perturbation on a gene means forcing the gene to take a fixed value. They implemented two methods to infer a genetic network built from a gene expression dataset. To implement the genetic network, they used a deterministic Boolean model. This model is a simplified version of Bayesian networks (see Section 3.1.1) where all variables are binary and all conditional distribution tables are simply truth tables.

Similar Boolean networks were used to model experiments involving gene networks, and a set-covering method was used to recommend the next best experiment [27]. Karp et al. used a Boolean circuit model of a biological pathway [2] to model experimentation.

## 4 Evaluation

This section describes an evaluation of the GEEVE system. In the evaluation, we used a simulator to generate gene expression data and compared the performance of the GEEVE system with a system of Tong and Koller [45] (call it the TK system) and a system of Ideker et al. [23] (call it the ID system), which are described in Section 2. Additionally, we compare GEEVE with a GEEVE base-line system that restricts GEEVE to consider a design protocol that contains only a single case (call it the GEEVE\_BL system).

### 4.1 Simulator for the evaluation

Only a few gene expression simulation systems are currently available [36, 37, 44]. Limited functions are available in most of the systems because they are in their early stages of development. For example, Tomita et al. [44] simulate a cell by developing a computer program shell that can execute any specified cell model. But the system is limited in its (1) available cell models, (2) only exporting simulated gene expression levels to a file, and (3) modeling of measurement errors.

We used the Scheines and Ramsey [37] simulator system (which we will call the SR Simulator) to generate gene expression data. The SR simulator models genes within a cell and incorporates biological variance, such as that due to signal loss or gene mutation, as well as measurement error. The simulator uses a user-defined number of cells in each probe (we set each probe to contain 100,000 cells in this study). It allows measurement at different time points and uses the following so called *Glass function* [10] to update an expression level of a gene  $X$ :

$$eX^t = eX^{t-1} + rate[-eX^{t-1} + F_X(\text{causes\_of}(X^t) \setminus X^{t-1})] + \varepsilon_X \quad (2)$$

where (1)  $X^t$  represents gene  $X$  at time  $t$ , (2)  $eX^t$  represents the expression level of gene  $X$  at time  $t$ , (3)  $0 < rate \leq 1$ , (4)  $\text{causes\_of}(X^t)$  are the gene regulators of  $X^t$  in the model, whereby each regulator is assumed to be either “on” or “off”, (5) “ $\setminus$ ” is the set difference operator, (6)  $\varepsilon_X$  is an error term drawn from a given probability distribution, and (7)  $F_X$  is a binary function specified by the user [10]. Binary functions have been used to model natural phenomena including gene causal pathways [28]. Also note that the model used in this evaluation study contains only a one-stage time-lag, an example of which is shown in Figure 7.

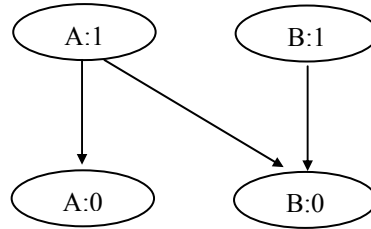


Figure 7. A one-stage time-lag model. A:0 represents the expression level of gene A at current time and A:1 represents the expression level of gene A at one time-step before the current time.

A burn-in period is desirable in applying the SR Simulator. In particular, for the simulated networks discussed in this section (1) it is often after 100 time lags that the most interesting interactions start among the modeled genes; and (2) the simulated system usually goes into a steady state after 300 time lags. Therefore we used 100 time lags for a burn-in period for the evaluation study reported here.

## 4.2 Simulated yeast galactose pathway

In the evaluation we used the SR simulator applied to the yeast galactose metabolic pathways [24] that includes nine galactose genes: *Gal1*, *Gal2*, *Gal3*, *Gal4*, *Gal5*, *Gal6*, *Gal7*, *Gal10*, and *Gal80*. The simulated causal pathway we used is shown in Figure 8; it only simulates the condition when galactose is provided as a nutrient. The causal pathway shown in Figure 8 was generated based on Ideker et al. [24].

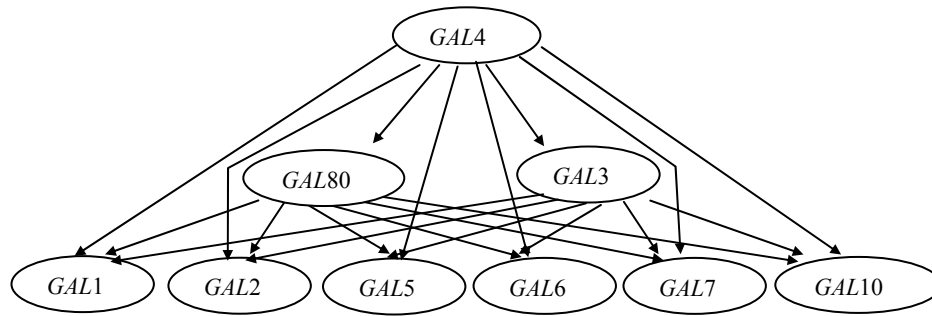


Figure 8. The galactose pathway model used in the simulation-based evaluation reported here.

The noise term  $\varepsilon_X$  in Equation 2 was estimated from the cDNA microarray dataset provided in Ideker et al. [24] and the *rate* parameter was estimated as 0.5 by a yeast biologist at our university.  $F_X$  in Equation 2 is defined in Table 1. The function was assessed based on Ideker et al. [24].

Table 1. Definition of  $F_X$  that appears in Equation 2. The cause is listed in the columns and the effect in the rows. 0 represents the gene is not expressed and 1 represents the gene is expressed. Examples of  $F_X$  are as follows (1) if *GAL4* is expressed then *GAL3* is suppressed and (2) if *GAL4* is not expressed then *GAL3* is expressed.

	<i>GAL4</i> =0	<i>GAL4</i> =1
<i>GAL3</i>	1	0

(a) *Gal4* and *Gal3*

	<i>GAL4</i> =0	<i>GAL4</i> =1
<i>GAL80</i>	0	1

(b) *Gal4* and *Gal80*

	$GAL4=0$				$GAL4=1$			
	$GAL3=0$		$GAL3=1$		$GAL3=0$		$GAL3=1$	
	$G80^*=0$	$G80^*=1$	$G80^*=0$	$G80^*=1$	$G80^*=0$	$G80^*=1$	$G80^*=0$	$G80^*=1$
$GO^\dagger$	0	0	0	0	1	1	0	0

(c) Other genes  $^\dagger GO = \{Gal1, Gal2, Gal5, Gal6, Gal7, Gal10\}$   $^* G80 \equiv Gal80$

### 4.3 Generated dataset

Initially, we simulated a possible the baseline study, where the experimenter collects an equal number of experimental repeats in different experimental conditions. A dataset of 30 *initial cases* were generated, three experimental repeats (cases) for each of the following 10 experimental conditions: a single wild-type experiment and nine knockout experiments, where a given knockout experiment corresponds to the deletion of one of the 9 genes in the simulator model. We generated these initial datasets (and subsequent ones) using a  $t=100$  burn-in period (see Section 4.1). After TK, ID, GEEVE\_BL, and GEEVE each analyzed the initial dataset of 30 cases, the following steps were iteratively taken with each system:

Step 1. The system outputs additional knockout experiments to perform.

Step 2. All of these experiments are performed (using the simulator).

Step 3. The system analyzes the results of the microarray experiments just performed (combined with the results of any earlier microarray experiments on the same genes under the same experimental conditions).

Step 4. If the total number of experiments performed thus far is 35 then halt; else go to Step 1.

Let  $D$  denote the dataset before Step 1. Then the dataset that is generated after Step 2 is  $D \cup \{\text{results of experiments that a system recommended}\}$ . Note that only GEEVE recommends more than one experiment to perform at a time.

The TK, ID, GEEVE\_BL, and GEEVE algorithms currently model using discrete variables only, although each could be extended to model with continuous variables as well. The following steps were taken for discretization (i.e., binning) of the simulated gene-expression data (generated from Equation 2) that were then used by the algorithms:

- (1) Let  $X_i^*$  denote the intensity for gene  $X_i$ , which serves as an indicator of the expression level of  $X_i$  in an experiment in which some gene (not necessarily  $X_i$ ) was knocked out. Similarly, let  $rX_i$  denote the intensity, which is an indicator of the expression level of  $X_i$  when no genes were manipulated (wild-type). The relative intensity for gene  $X_i$  was calculated as  $\log(X_i^*/rX_i^*)$ .
- (2) Discretization was performed based on each gene's relative intensity of mean  $m$  and standard deviation  $\delta$  over all relative intensities. All genes were assigned three states: 0 was assigned to any value less than  $m-\delta$ , 1 was assigned to any value greater than or equal to  $m-\delta$  and less than  $m+\delta$ , and 2 was assigned to any value greater than or equal to  $m+\delta$ .

For the discretization for the ID system (Step b), which assumes binary variables, all genes were assigned two states: 0 was assigned to any value less than  $m$ , and 1 was assigned to any value greater than or equal to  $m$ .

#### 4.4 Evaluation matrices

TK and ID do not incorporate experimenter's preferences. To make the comparison of TK and ID with GEEVE fair, in this comparison study we use uniform preference on all gene pairs and

causal hypotheses, as described in Section 2.2. Since TK and ID do not model latent variable, we did not consider latent confounded relationships, i.e., relationships between  $X$  and  $Y$  in Figure 3 are grouped as (1) causally independent for  $E_3$  or  $E_6$ ; and (2) causally related for  $E_1, E_2, E_4$ , or  $E_5$ . The measurement of the performance of TK, ID, GEEVE\_BL and GEEVE was based on the two criteria discussed next.

1) **Prediction of the generating causal relationship.** Using each system's prediction scores of all pairwise causal relationships among all 9 yeast galactose genes, we calculated two AUROCs under two different categories, i.e., "Independence Prediction" and "Causal Prediction". In "Independence Prediction" category, we show how well each system predicts independence relationships. Thus in this category,  $E_3$  is the true state and  $\{E_1, E_2\}$  are the false state that each system has to predict. In "Causal Prediction" category, we show how well each system predicts causal relationships. Thus in this category, (1) if  $E_1$  is the true state then  $\{E_2, E_3\}$  are the false state; or (2) if  $E_2$  is the true state then  $\{E_1, E_3\}$  are the false state that each system has to predict. For each algorithm and for each number of experiments performed we calculate two types of AUROC.

2) **Predictive performance as a function of the number of experiments performed.** Using the cost function that was estimated by a yeast researcher (detail function descriptions are in Section 4.5), we calculated the total cost of the experiments performed by each system. For example, if it takes 2 hours to process a microarray chip and it costs \$50.00/hour for such analysis, the total cost is \$100.00 (excluding the material costs, which we assume are fixed per microarray experiment). As mentioned above, predictive performance is measured using

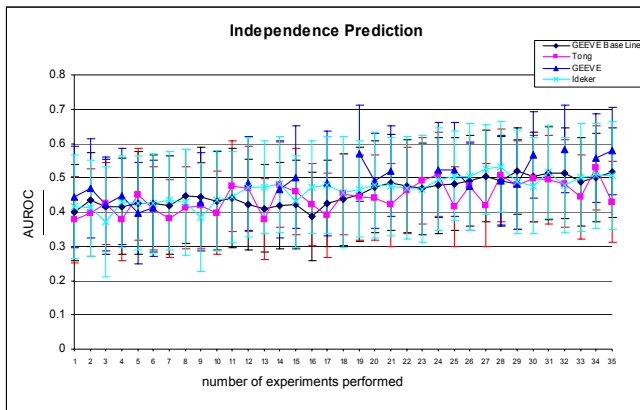
AUROC, which is derived by using the generating relationships as the ground truth. Finally, using the assessed cost function in Section 4.5, we recorded the cost that is associated with attaining a given AUROC and use these factors to derive a unit “*performance accuracy per cost*,” which is calculated by dividing the AUROC by the experiment costs in dollars. This unit represents an increased fraction of an AUROC per dollar cost. We plot the AUROC over cost as a function of the number of experiments performed.

In analyzing a given set of data with a given system, we ran the system for up to two hours for the following reason. A running time of less than two hours showed relatively high variance on AUROC to the variance on AUROC for a running time of over two hours. Furthermore, a running time of three hours to four hours showed similar variance on AUROC to that of the two-hour running time. We used a 500MHz dual processor Linux machine to set up the appropriate parameters for each system to run approximately two hours. For the entire experiment, we used the Linux machine, a 400MHz Microsoft Windows 2000 machine, and a 266MHz Microsoft Windows NT machine. All programs were compiled with gnu C++ on the Linux machine and with Microsoft Visual C++ on the Windows machines. The total running time was approximately as follows: 35 experiments  $\times$  2 hours per system  $\times$  4 systems (i.e., TK, ID, GEEVE, and GEEVE\_BL) per experiment = 280 hours.

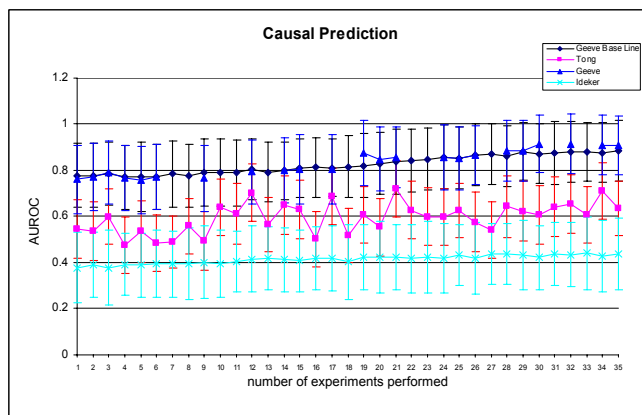
## 4.5 Results

Results of the predictive performance of each system are shown in Figure 9(a) and Figure 9(b). The  $X$ -axis represents the number of experiments performed (using simulation) and analyzed by the system. As shown in Figure 9(a), there is no system that dominates in predicting the correct

independence relationships. In contrast, the results in Figure 9(b) show that GEEVE and GEEVE\_BL performed better on average than did TK and ID.



(a) AUROC of independence relationships prediction



(b) AUROC of causal relationships prediction

Figure 9. Area under ROC curve (AUROC) as a function of the number of experimental cases performed (via simulation) and used to assess relationships among the variables. Each bar represents a 95% confidence interval.

Recall that except for GEEVE, all other systems recommend one microarray experiment at a time. This is why the GEEVE plots are disconnected in Figure 9. For example, in Figure 9, GEEVE requests two microarray experiments after it analyzes 15 microarray experiments; after analyzing 17 microarray experiments, it again requests two microarray experiments. Error bars of the AUROC in the figure were calculated using the bootstrap method described in Efron [11]. In particular, for an AUROC curve for a given system, that systems 36 predictions were randomly selected with replacement and this procedure was performed 2,000 times. The error bars each represent a 95% confidence interval around a given AUROC.

We also note that Tong and Koller compared the TK system with a system that uses Bayesian networks and determines the next experiment uniformly at random, which constitutes a type of

random-experiment *baseline system* [45]. They showed that TK outperforms the baseline system in terms of predicting the generating structures. Thus, we expect the TK system in Figure 9 would perform at least as well as such a baseline system in terms of AUROC.

Note that we have to consider the cost to process a cDNA microarray chip. A large portion of the cost is the technician's time to process the microarray chip. Processing one microarray chip at a time is much more costly than batch-processing several microarray chips at a time. This is because there are many steps to take to analyze a cDNA microarray chip, and each step takes a long time to complete. Consider the following scenario for a given experiment that involves a microarray: (1) perform experiment  $\xi$ , (2) analyze the microarray chip results of experiment  $\xi$ ; (3) based on the results, determine the next experiment  $\xi'$  to perform; (4) perform the experiment  $\xi'$  with a microarray chip; and (5) analyze the microarray chip results of experiment  $\xi'$ . In this scenario, a technician analyzed two microarray chips in all. Now consider the alternate scenario where the technician performs experiments  $\xi$  and  $\xi'$  together and analyzes the resulting two microarray chips in a batch mode.

Typically, analyzing two chips together will take less of the technician's time than analyzing two chips in series, particularly if  $\xi = \xi'$ , that is,  $\xi'$  is simply a repeat of  $\xi$ . The downside, however, is that in doing two experiments at the same time, we are not able to use the results of the first experiment to help tailor which experiment to perform second.

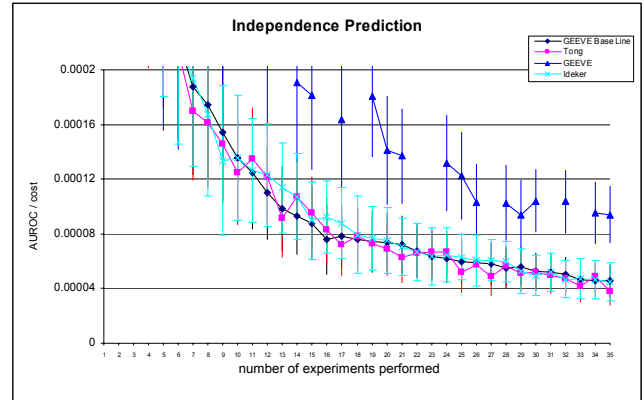
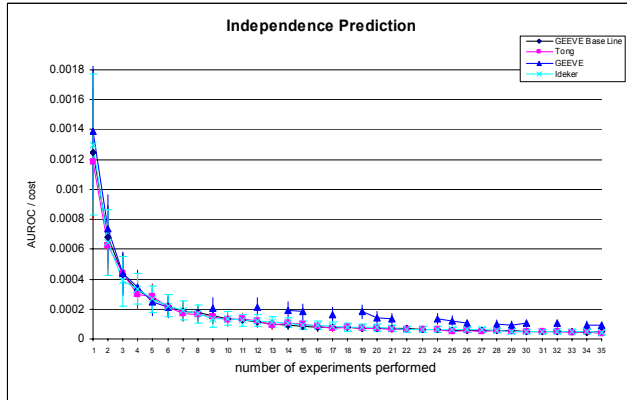
There are different suggested protocols to analyze a microarray chip [20]. We estimate that currently it takes about 16 hours (two work days) of a technician's time to produce and analyze one microarray chip. It will usually take 20 and 24 hours for him or her to analyze two and three

cDNA microarray chips at once, respectively (four hours for each additional microarray chip).

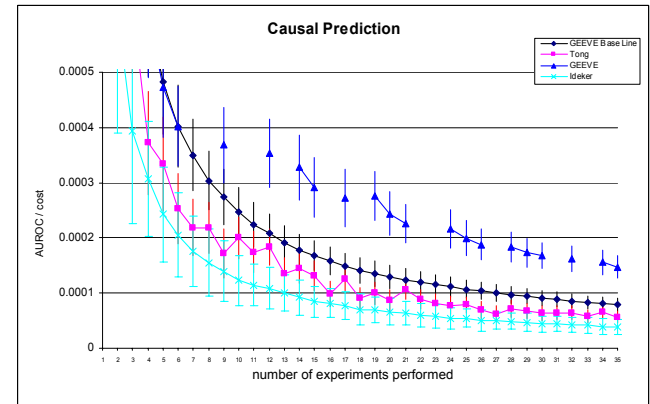
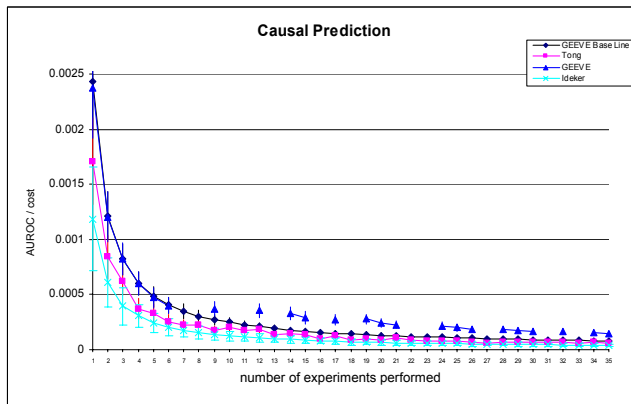
This is because it usually takes four hours to finish the first step, extracting DNA. If the technician earns \$20 per hour, the costs involved in analyzing two chips in the two different scenarios are: (1) \$640 to analyze one chip at a time  $[(16 \text{ hours} \times 2) \times \$20]$ ; and (2) \$400 to analyze two chips at once  $(20 \text{ hours} \times \$20)$ . Similarly, the costs involved in analyzing three chips are: (1) \$960 to analyze one chip at a time  $[(16 \text{ hours} \times 3) \times \$20]$ ; and (2) \$480 to analyze three chips at once  $(24 \text{ hours} \times \$20)$ . We used these cost assumptions in the analyses that follow.

Under these cost assumptions, we can calculate AUROC per dollar, which is shown in Figure 10. It is clear that under these cost assumptions, GEEVE outperforms the other systems in both causal and independence predictions.

In summary, Figure 9(b) shows that GEEVE and GEEVE\_BL consistently performed better than the ID and TK systems in correctly predicting causal relationships. Figure 10(b) shows that GEEVE and GEEVE\_BL outperform the ID and TK systems in predictive performance per dollar. GEEVE outperforms GEEVE\_BL in Figure 10(b) because GEEVE recommends more than one microarray experiment (case) at a time.



(a) AUROC/cost of independence relationships prediction. The left graph shows the overall plot and the right graph shows the left graph's lower right plot in detail.



(b) AUROC/cost of causal relationships prediction. The left graph shows the overall plot and the right graph shows the left graph's lower right plot in detail.

Figure 10. AUROC per cost calculation. Each bar represents a 95% confidence interval. The X axis represents the number of microarray experiments that were suggested by an algorithm and then performed by way of simulation.

## 5 Conclusions

Systems biology emphasizes large scale discovery of the *interactions* of genes, proteins, and other cell elements. Systems biology is confronted with a huge number of interactions, not the least of which is the interaction of genes. There are challenges in designing high throughput experiments, such as cDNA microarrays, and for analyzing the high volume of data generated by those experiments in order to discover gene regulation networks. Intrinsically, these issues are

causal in nature. We have introduced a new causal analysis method along with a computer system that uses that method to recommend the gene-regulation experiments to perform.

Unlike clinical randomized controlled trials, where an experimenter is interested in the causal relationship of a handful variables (e.g., an experimenter is interested in a new drug and its treatment effect) in systems biology an experimenter is usually interested in the causal relationships among thousands of entities, such as genes. Different approaches are needed in systems biology for causal discovery and experimental design recommendation. This paper has explored one such approach. In the remainder of this section, we summarize the contributions made by this paper and then discuss open problems.

## **5.1 Local causal search with experimentation recommendations**

We developed a system called GEEVE (causal discovery in Gene Expression data using Expected Value of Experimentation) that incorporates an experimenter's preferences regarding which genes to study in order to discover causal relationships among those genes. Among the genes of interest, GEEVE models their likely causal relationships, based on prior biological knowledge and experimental data.

Experiments provide benefit in terms of information, but they also have costs in terms of human labor and the laboratory costs. Considering preferences, costs, and a current model of causal relationships, GEEVE recommends the most cost-effective experiment it can find in its search of the space of experiments.

For evaluation, we modeled and simulated a portion of the yeast galactose metabolic pathway. Using the yeast galactose pathway simulator to generate simulated microarray data, we showed that GEEVE predictions (area under ROC curve) were better (although not highly statistically significantly so) than two other state-of-the-art methods recently described in the literature. When we applied a cost function and calculated the area under ROC curve as a function of experimental cost, GEEVE showed performance that was statistically significantly different than the other two recommendation systems.

## **5.2 Future work and open issues**

External experimental conditions, such as nutrient conditions, could be modeled as exogenous variables in a causal Bayesian network. Currently, GEEVE (and LIM) models only experiments that involve wild-type gene levels and single gene knock-outs. In the future, more general experiments, such as over-expression experiments, more than one gene knock-out and so forth, should be modeled. Microarray datasets under different interventions are becoming more readily available, e.g., yeast knockout experiments datasets [14]. Our immediate interest is to apply LIM to these actual biological datasets.

Regarding modeling the time course of gene expression, and determining precisely when to sample cells during experimentation, temporal Bayesian networks appear a natural choice [13, 33]. It will be interesting to explore models that use both continuous and discrete variables within temporal Bayesian networks. Temporal Bayesian networks also provide one approach to modeling gene regulation feedback. The six pairwise causal hypotheses used in this research could be extended to model such feedback. This is an important issue for future research because feedback is widely observed in many cellular pathways.

Currently GEEVE only generates decision trees based on the discovery of pairwise gene relationships. More generally,  $R_j$  in Figure 5 (Section 2.3) should include more than pairwise relationships. Doing so will allow GEEVE to (1) model beyond a single gene perturbation experiments, such as a knockout of two or more genes at a time; and (2) incorporate (in the decision tree) the effects on other genes besides genes ( $X$ ,  $Y$ ) when gene  $X$  (or  $Y$ ) is perturbed.

We have also introduced a causal discovery system that can score latent structures. Since the most closely related prior methods assume no latent variables, there is no straightforward way to evaluate GEEVE's prediction of latent structures with these other methods. Also since a cDNA microarray measures the average expression level of millions of cells, the variance that we observe in the levels (when an experiment is repeated several times) is due almost entirely to measurement error and not to biological variation [43]. Biological variation is needed to discover latent structure, certainly with LIM, and we believe with any method. Measuring the expression level of genes under various experimental conditions (e.g., measuring at different time points or at different temperatures) can provide biological variation among groups of cells; it is an open question how helpful biological variation of this particular variety will be in discovery of latent structure.

Another way to obtain biological variation in gene expression would be to measure gene expression at the level of a single cell. Such measurements will require new technology. We anticipate that such methods will be developed within the next decade. If so, the methods in this

paper will be applicable to suggesting when latent factors (such as unknown proteins) may be influencing two or more specific genes.

LIM searches over local causal structures and constructs the most probable pairs of causal relationships. LIM makes no distinction between direct and indirect causation, relative to the variables being modeled. That is, if LIM outputs that  $A$  causes  $C$ , it could be that  $A$  does so through  $B$  (and perhaps even only through  $B$ ) where all of  $A$ ,  $B$ , and  $C$  are being modeled. It would be useful to integrate LIM's local causal search results in order to construct a global causal network that contains only direct causal relationships.

Ideker et al. [24] describe four steps in discovering causal pathways among the genes: (1) gather and formulate the current knowledge about the genes and their pathways; (2) design and perform experiments; (3) analyze the data from the experiments; and (4) formulate new hypotheses to explain the analysis results not predicted by Step 1 and then repeat steps 2, 3, and 4. There are many open issues in how to complete this loop. The soundness of microarray measurements needs to be studied further, e.g., studying the relationship between mRNA levels and protein expression levels, and studying and quantifying the various sources of measurement error related to detecting gene expression levels. Other open issues include detecting genes and their promoter regions from sequence information, compiling known gene regulatory knowledge (and other cell-network knowledge) from the literature, and then using this prior knowledge combined with microarray data to derive posterior probabilities over causal hypotheses about gene regulation.

## Acknowledgements

We thank the Computational Systems Biology Group

(<http://www.phil.cmu.edu/projects/genegroup/>) and Professor Martin Shmidit for helpful comments in the early development of GEEVE. This research has been partially supported by grants from the National Aeronautics and Space Administration (NRA2-37143) the National Science Foundation (IIS-9812021), the Mellon Fellowship of University of Pittsburgh.

## Bibliography

1. Achcar, J.A., *Use of Bayesian analysis to design of clinical trials with one treatment*. Communications in Statistics, Theory, and Methods, 1984. **13**: p. 1693-1707.
2. Akutsu, T., S. Miyano, and S. Kuhara. *Identification of genetic networks from a small number of gene expression patterns under the Boolean network model*. in *Pacific Symposium on Biocomputing*. 1999. Hawaii p. 17-28.
3. Berry, D.A. and D.K. Stangl, *Bayesian methods in health-related research*, in *Bayesian Biostatistics*, D.A. Berry and D.K. Stangl, Editors. 1996, Marcel Dekker: New York. p. 3-66.
4. Brown, P.O. and D. Botstein, *Exploring the new world of the genome with DNA microarrays*. Nature Genetics, 1999. **21**(supplement): p. 33-37.
5. Chavez, T. and M. Henrion. *Efficient estimation of the value of information in Monte Carlo models*. in *Uncertainty in Artificial Intelligence*. 1994 p. 119-127.
6. Cooper, G.F. and E. Herskovits, *A Bayesian method for the induction of probabilistic networks from data*. Machine Learning, 1992. **9**: p. 309-347.
7. Cooper, G.F. and C. Yoo. *Causal discovery from a mixture of experimental and observational data*. in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. 1999: Morgan Kaufmann p. 116-125.
8. de Jong, H., *Modeling and simulation of genetic regulatory systems: a literature review*. Journal of computational biology, 2002. **9**(1): p. 67-103.
9. Dutilh, B. *Gene Networks from Microarray Data*. in *Unpublished manuscript*. 1999. Literature thesis at Utrecht University
10. Edwards, R. and L. Glass, *Combinatorial explosion in model gene networks*. Chaos, 2000. **10**: p. 691-704.
11. Efron, B., *The jackknife, the bootstrap and other resampling plans*. Society for the Industrial and Applied Mathematics, 1982(1092): p. 2-5.
12. Friedman, L.M., C.D. Furberg, and D.L. DeMets, *Chapter 7. Sample size*, in *Fundamentals of clinical trials, 3rd Edition*, 94-129, Editor. 1996, Mosby-Year book. p. St. Louis.
13. Friedman, N., et al., *Using Bayesian networks to analyze expression data*. Journal of Computational Biology, 2000.
14. Giaever, G., et al., *Functional Profiling of the Saccharomyces cerevisiae Genome*. Nature, 2002. **418**: p. 387-391.

15. Golub, T.R., et al., *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*. Science, 1999. **286**: p. 531-537.
16. Heckerman, D., E. Horvitz, and B. Middleton. *An approximate nonmyopic computation for value of information*. in *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*. 1991
17. Heckerman, D. *A Bayesian approach to learning causal networks*. in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. 1995: Morgan Kaufmann p. 285-295.
18. Heckerman, D., D. Geiger, and D. Chickering, *Learning Bayesian networks: The combination of knowledge and statistical data*. Machine Learning, 1995. **20**: p. 197-243.
19. Heckerman, D., C. Meek, and G.F. Cooper, *A Bayesian approach to causal discovery*, in *Computation, Causation, and Discovery*, C. Glymour and G.F. Cooper, Editors. 1999, AAAI Press: Menlo Park, CA. p. 141-165.
20. Hegde, P., et al., *A Concise Guide to cDNA Microarray Analysis*. Biotechniques, 2000. **29**(3): p. 548-562.
21. Henrion, M., *Propagating uncertainty in Bayesian networks by probabilistic logic sampling*, in *Uncertainty in Artificial Intelligence 2*, J.F. Lemmer and L.N. Kanal, Editors. 1988, North-Holland: Amsterdam. p. 149-163.
22. Herwig, R., et al., *Large-scale clustering of cDNA-fingerprinting data*. Genome Research, 1999. **9**: p. 1093-1105.
23. Ideker, T., V. Thorsson, and R.M. Karp. *Discovery of regulatory interactions through perturbation: inference and experimental design*. in *Pacific Symposium Biocomputation*. 2000 p. 305-316.
24. Ideker, T., et al., *Integrated genomic and proteomic analysis of a systematically perturbed metabolic network*. Science, 2001. **292**: p. 929-934.
25. Karp, P.D., *Hypothesis Formation as Design*, in *Computational Models of Discovery and Theory Formation*, J. Shrager and P. Langley, Editors. 1990, Morgan Kaufman: San Mateo, CA. p. 276-317.
26. Karp, P.D., et al., *Integrated pathway/ genome database and their role in drug discovery*. Trends in Biotechnology, 1999. **17**(7): p. 275-281.
27. Karp, R.M., R. Stoughton, and K.Y. Yeung. *Algorithms for choosing differential gene expression experiments*. in *Research in Computational Biology*. 1999 p. 208-217.
28. Kauffman, S., *Origins of Order - Self-Organization and Selection in Evolution*. 1993: Oxford University Press.
29. Keeney, R.L. and H. Raiffa, *Decisions with multiple objectives: preference and value tradeoffs*. 1976, New York: John Wiley.
30. Kerr, M.K. and G.A. Churchill, *Experimental design for gene expression microarrays*. Biostatistics, 2001. **2**: p. 183-201.
31. Lipshutz, R.J., et al., *High density synthetic oligonucleotide arrays*. Nature Genetics, 1999. **21**(supplement): p. 20-24.
32. Michaels, G.S., et al. *Cluster analysis and data visualization of large-scale gene expression data*. in *Pacific Symposium on Biocomputing*. 1998 p. 42-53.
33. Murphy, K. and S. Mian, *Modelling Gene Expression Data using Dynamic Bayesian Networks*, in *Technical report*, U.B. Department of Computer Science, Editor. 1999.
34. Pearl, J., *Probabilistic Reasoning in Intelligent Systems*. Representation and Reasoning, ed. R.J. Brachman. 1988, San Mateo, CA: Morgan Kaufmann.

35. Pearl, J., *Causality: Models, Reasoning, and Inference*. 2000, Cambridge, UK: Cambridge University Press.
36. Saavedra, R. and C. Glymour, *A Regulatory Network Simulator*, in *Simulator based on (Yuh et al. 1998) under development*. 2001.
37. Scheines, R. and J. Ramsey, *Gene simulator*, in *Available at: <http://www.phil.cmu.edu/tetrad/>*. 2001.
38. Shrager, J. and P. Langley, *Computational Models of Discovery and Theory Formation*, ed. J. Shrager and P. Langley. 1990, San Mateo, CA: Morgan Kaufman.
39. Smolen, P., D.A. Baxter, and J.H. Byrne, *Modeling transcriptional control in gene networks - methods, recent results and future directions*. *Bulletin of Mathematical Biology*, 2000. **62**: p. 247-292.
40. Spellman, P.T., et al., *Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization*. *Molecular Biology of the Cell*, 1998. **9**: p. 3273-3297.
41. Spiegelhalter, D.J., L.S. Freedman, and M.K.B. Parmar, *Bayesian approach to randomized trials*. *Journal of the Royal Statistical Society*, 1994. **157**(Part 3): p. 357-416.
42. Spirtes, P., C. Glymour, and R. Scheines, *Causation, prediction, and search*. 2 ed. 2000, Cambridge, MA: MIT Press.
43. Spirtes, P., C. Glymour, and R. Scheines. *Constructing Bayesian network models of gene expression networks from microarray data*. in *to appear in the Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology*. 2001
44. Tomita, M., et al., *E-CELL: Software environment for whole cell simulation*. *Bioinformatics*, 1999. **15**(1): p. 72-84.
45. Tong, S. and D. Koller. *Active learning for structure in Bayesian networks*. in *International Joint Conference on Artificial Intelligence*. 2001. Seattle WA
46. Tsang, J., *Gene expression, DNA arrays, and genetic network*, in *Unpublished manuscript Bioinformatics Laboratory at University of Waterloo*. 1999.
47. von Neumann, J. and O. Morgenstern, *Theory of Games and Economic Behavior*. 1944, Princeton Univ. Press: Princeton NJ.
48. Yoo, C. and G. Cooper, *Causal discovery of latent-variable models from a mixture of experimental and observational data*, in *Center for Biomedical Informatics Research Report CBMI-173*. 2001, Center for Biomedical Informatics: Pittsburgh, PA.
49. Yoo, C., *Expected Value of Experimentation in Causal Discovery from Gene Expression Studies*. Ph.D. dissertation, 2002.
50. Yoo, C. and G. Cooper. *Discovery of gene-regulation pathways using local causal search*. in *AMIA*. 2002. San Antonio, Texas p. 914-918.
51. Yoo, C., V. Thorsson, and G.F. Cooper. *Discovery of a gene-regulation pathway from a mixture of experimental and observational DNA microarray data*. in *Pacific Symposium on Biocomputing*. 2002. Maui, Hawaii: World Scientific p. 498-509.