Local Causal Discovery Algorithm using Causal Bayesian networks

Changwon Yoo
University of Montana
Missoula, MT 59806
cwyoo@cs.umt.edu

Erik M. Brilz
University of Montana
Missoula, MT 59806
brillo@montana.com

Abstract

To learn about the progression of a complex disease, it is necessary to understand the physiology and function of many genes operating together in distinct interactions as a system. In order to significantly advance our understanding of the function of a system, we need to learn the causal relationships among its modeled genes. To this end, it is desirable to compare experiments of the system under complete interventions of some genes, e.g., knock-out of some genes, with experiments of the system under no interventions. However, it is expensive and difficult (if not impossible) to conduct wet lab experiments of complete interventions of genes in animal models, e.g., a mouse model. Thus, it will be helpful if we can discover promising causal relationships among genes with observational data alone in order to identify promising genes to perturb in the system that can later be verified in wet laboratories. While causal Bayesian networks have been actively used in discovering gene pathways, most of the algorithms that discover pairwise causal relationships from observational data alone identify only a small number of significant pairwise causal relationships, even with a large dataset. In this paper we introduce a new causal discovery algorithm—the Equivalence Local Implicit latent variable scoring Method (EquLIM)—that identifies promising causal relationships even with a small observational dataset. We also compare the prediction performance of EquLIM using area under receiver operating characteristics (AUROC) curve on data generated from a gene network simulator. EquLIM showed better prediction performance than LIM, which uses a greedy hill climbing Bayesian network structure search.

1 INTRODUCTION

Causal modeling and discovery are fundamental pursuits of science. Experimental studies, such as randomized controlled trials (RCTs), often provide the most trustworthy methods we have for establishing causal relationships from data. Observational data are passively observed. Such data are more readily available than are experimental data. As observational electronic databases become increasingly available, the opportunities for using them for causal discovery also increase. In an experimental study, one or more variables are manipulated (typically randomly) and the effects on other variables are measured. Such studies, while potentially very informative, may be expensive and difficult (if not impossible) to conduct wet lab experiments of complete interventions of genes in animal models, e.g., a mouse model. Thus, it will be helpful if we can discover promising causal relationships among genes with observational data alone in order to identify promising genes to perturb in the system that can later be verified in wet laboratories.

Bayesian discovery of causal networks is an active field of research in which numerous advances have been — and continue to be — made in areas that include causal representation, model assessment and scoring, and model search (Spirtes, Glymour et al. 2000). In prior work on Bayesian discovery of causal networks, researchers have focused primarily on methods for discovering causal relationships from observational data (Heckerman, Geiger et al. 1995; Spirtes, Glymour et al. 2000). A notable exception is a paper by Heckerman on learning influence diagrams as causal models. It contains key ideas for learning causal Bayesian networks from a combination of both experimental data under deterministic manipulation and observational data (Heckerman 1995).

The contribution of the current paper is to introduce and investigate a new causal discovery algorithm that identifies promising causal relationships even with a small observational dataset. It extends the earlier work (Cooper and Yoo 1999) by improving the search methods to discover promising causal relationships on observational data alone. We describe Bayesian methods for learning Bayesian networks when variable manipulation is not necessarily deterministic, but rather stochastic. In the important special case in which manipulation is deterministic, there is a closed-form Bayesian scoring metric that is a simple variation on a previous scoring metric for Bayesian network learning (Cooper and Herskovits 1992; Heckerman, Geiger et al. 1995). We introduce a novel pairwise relationship scoring method — Equivalence Local Implicit latent variable scoring Method (EquLIM) — to learn causal networks from observational data alone. Next we investigate the learning performance, using area under receiver operating characteristics (AUROC) curve of EquLIM. This evaluation is a simulation study in which cases were generated from a gene network simulator.

2 MODELING METHOD

A Bayesian network is a directed acyclic graph in which each node represents a variable and each arc represents probabilistic influence. 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 (Pearl 1988). Figure 1 illustrates the structure of a hypothetical causal Bayesian network structure that contains five nodes. The probabilities associated with this causal network structure are not shown.

The causal network structure in Figure 1 indicates, for example, that the *Gene*1 can regulate (causally influence) the expression level of the *Gene*3 gene, which in turn can regulate the expression level of *Gene*5 gene.

The causal Markov condition gives the conditional independence relationships that are specified by a causal Bayesian network:

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

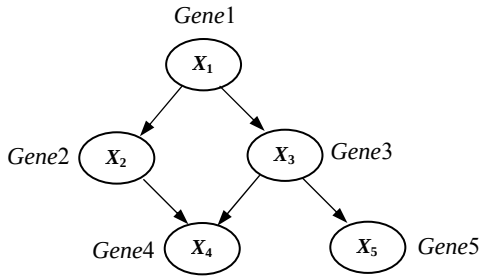

Figure 1. A causal Bayesian network that represents a hypothetical gene-regulation pathway.

The causal Markov condition permits the joint distribution of the n variables in a causal Bayesian network to be factored as follows(Pearl 1988):

$$P(x_1, x_2, ..., x_n \mid K) = \prod_{i=1}^{n} P(x_i \mid \pi_i, K) \qquad (1)$$

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

2.1 STRUCTURE LEARNING

We introduce six equivalence classes (E_1 through E_6) among the structures (Figure 2). The causal networks in an equivalence class are statistically indistinguishable for any observational and experimental data on X and Y. We denote an arbitrary pair of nodes in a given Bayesian network B as (X, Y). If there is at least one directed causal path from X to Y or from Y to X, we say that X and Y are *causally related* in B. If X and Y share a common ancestor, we say that X and Y are *confounded* in B. As the first step toward Bayesian casual modeling of latent variables, which is a computationally challenging problem, we only look at pairwise relationships between two nodes (X and Y) and a latent variable H.

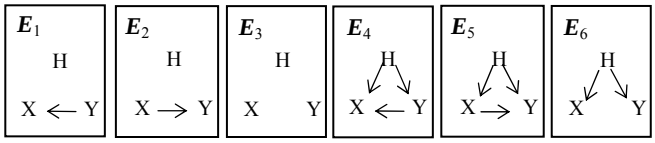

Figure 2. Six Local Causal Hypotheses

Let $E = \{E_1, E_2, E_3, E_4, E_5, E_6\}$ and let E_i^{XY} denote the node pair X and Y with causal relationship E_i. Let us consider the posterior probability that variable X causes variable Y given data D on the measured variables. We can derive the posterior probability of E_i^{XY} as:

$$P(E_i^{XY} \mid D, K) = \sum_{S: E_i^{XY} \text{ is in } S} P(S \mid D, K) \qquad (2)$$

where the sum is taken over all admissible causal network structures S, such that S contains substructure E_i^{XY}. An *admissible causal network structure* is a structure that (1) contains all of the variables being modeled (of which X and Y are but two), and (2) has a non-zero prior probability. Based on the properties of probabilities, the term within the sum in Equation 2 may be rewritten as follows:

$$P(S \mid D, K) = \frac{P(S, D \mid K)}{P(D \mid K)} = \frac{P(S, D \mid K)}{\sum_{S} P(S, D \mid K)} \qquad (3)$$

Since the probability $P(D \mid K)$ is a constant relative to the entire set of causal structures being considered, Equation 3 shows that the posterior probability of causal structure S is proportional to $P(S, D \mid K)$, which we can view as a *score* of S in the context of D. The probability terms on the right side of Equation 3 may be expanded as follows:

$$\begin{aligned} P(S, D \mid K) &= P(S \mid K) P(D \mid S, K) \\ &= P(S \mid K) \int P(D \mid S, \theta_S, K) P(\theta_S \mid S, K) d\theta_S \end{aligned} \qquad (4)$$

where (1) $P(S \mid K)$ is a prior belief that network structure S correctly captures the qualitative causal relationships among all the modeled variables; (2) θ_S are the probabilities (parameters) that relate the nodes in S quantitatively to their respective parents; (3) $P(D \mid S, \theta_S, K)$ is the likelihood of data D being produced, given that the causal process generating the data is a causal Bayesian network given by S and θ_S; and (4) $P(\theta_S \mid S, K)$ expresses a prior belief about the probability distributions that serve to model the underlying causal process.

With appropriate assumptions, we can evaluate $P(D|S,K)$ in Equation 4 with the following equation (Cooper and Herskovits 1992; Heckerman, Geiger et al. 1995):

$$P(D \mid S, K) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \qquad (5)$$

where r_i is the number of states that X_i can have, q_i denotes the number of joint states that the parents of X_i can have, N_{ijk} is the number of cases in D in which node X_i is *passively observed* to have state k when its parents have states as given by j, Γ is the gamma function, α_{ijk} and α_{ij} express parameters of the Dirichlet prior distributions, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. We used the BDe metric (Heckerman, Geiger et al. 1995) with $\alpha_{ijk} = \frac{1}{r_i q_i}$, which is a commonly used non-informative parameter prior for the BDe metric.

2.2 MODELING MANIPULATION

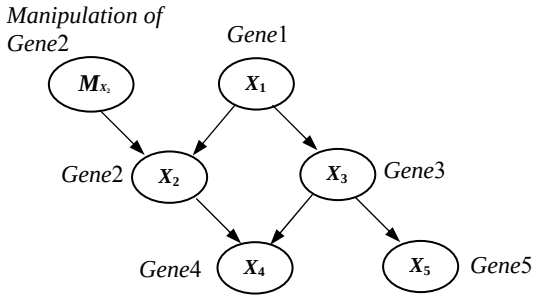

Figure 3. A hypothetical gene-regulation pathway with manipulation.

In the current section, we consider the situation in which manipulation of a variable may not be deterministic. A classic example from medicine is when a patient, who has volunteered to participate in a study, is randomized to receive a certain medication, but he or she decides not to take it. Let M_{X_i} be a variable that represents the value k (from 1 to r_i) to which the experimenter wishes to manipulate X_i. Let $M_{X_i} = o$ denote that the experimenter does not wish to manipulate X_i, but merely wants to observe its value. Augment the model variables to include M_{X_i}. Finally, carry out the analysis in section 2.1 assuming only observational data. The causal network hypotheses used in that analysis will include probabilities that specify prior beliefs about the causal influence of M_{X_i} on X_i. Those prior beliefs (on structure and parameters) will be updated by data on stated experimental intentions and observed variable value outcomes. For the special case of deterministic manipulation, we have that (1) with probability 1 variable M_{X_i} is a parent of X_i; and (2) $P(X_i = k | M_{X_i} = k, \pi_i') = 1$, where π_i' are the parents of X_i other than M_{X_i}. When scoring X_i, deterministic manipulation is equivalent to ignoring the cases in which X_i was manipulated (Cooper and Yoo 1999). In particular, to incorporate experimental data, we evaluate Equation 5 by not adding the cases to N_{ijk} when X_i is manipulated (Cooper and Yoo 1999). In the remainder of this paper, we assume that manipulation is deterministic.

For example, Figure 3 shows a causal network structure that has an additional variable M_{X_2} relative to the network structure in Figure 1. In Figure 3, variable X_2 (*Gene*2) is modeled as being manipulated in some cases, such as, knocking out *Gene*2.

3 EQUIVALANCE LOCAL IMPLICIT LATENT VARIABLE SCORING METHOD

In this section we introduce the implicit latent variable scoring (ILVS) method and then introduce a method called Local ILVS Method (LIM) that extends ILVS. At the end we introduce Equivalence LIM (EquLIM).

We denote $D(X, Y)$ as a set of cases (in dataset D) in which both node X and Y are observed. $D(mX, Y)$ denotes a set of cases in which node X is manipulated and Y is observed. Similarly, $D(X, mY)$ denotes a set in which node X is observed and Y is manipulated.

Implicit Latent Variable Scoring (ILVS) Method. Explicit scoring of latent-variable models requires exponential time in the number of database samples. Therefore, approximation methods have been introduced in the literature, including methods based on stochastic simulation and on expectation maximization(Heckerman, Geiger et al. 1995). Unfortunately, these methods often require long computation times before producing acceptable approximations. Therefore, we developed a new method called the Implicit Latent Variable Scoring (ILVS) method(Yoo and Cooper 2001).

The basic idea underlying ILVS is to (1) transform the scoring of a latent model E_i (e.g., E_5 in Figure 2) into the scoring of multiple non-latent variable models, (2) score those non-latent models efficiently using Equation 2, and then (3) combine the results of those scores to derive an overall score (i.e., marginal likelihood). For instance, consider scoring E_5 with two types of samples. One type is data for which X and Y were passively observed. We can derive the marginal likelihood of this data using the causal network in Figure 3(a), which contains no latent variable. Let $P(D_o \mid E_5, K)$ denote this marginal likelihood. The other type of sample is data for which X was manipulated and Y was observed. We use the causal network in Figure 4 (b) to derive the marginal likelihood of this data, namely $P(D_m \mid E_5, K)$. The different appearance of the arcs in Figure 4 (a) and Figure 4(b) signifies that these arcs are representing different distributions of X and Y. Continuing the Bayesian analysis, if (as in ILVS) we assume our beliefs about the distribution of X and Y in the Figure 3(a) situation are independent of the beliefs about their distribution in the Figure 4(b) situation, then the overall marginal likelihood of all the data (the passively observed data and the data generated by experimental manipulation) is $P(D \mid E_5, K) = P(D_o \mid E_5, K) \times P(D_m \mid E_5, K)$. It is straightforward to

extend the analysis to also include data in which Y was manipulated and X was passively observed.

In deriving the marginal likelihood of E_4 and E_6, ILVS uses a technique similar to the one just described for E_5. Yoo and Cooper(2001) provide algorithmic details of ILVS and a proof of its convergence to the correct generating structure in the large sample limit.

$X \Longrightarrow Y$ (a) $X \longrightarrow Y$ (b)

Figure 4. Two non-latent variable structures used to score a latent-variable structure.

ILVS scores each E_i in Figure 2 by only considering pairwise measured nodes. Thus, ILVS evaluates Equation 5 for only two measured nodes at a time. In earlier studies, we applied ILVS to simulated data(Yoo and Cooper 2001) and to yeast DNA microarray data(Yoo, Thorsson et al. 2002). We have also extended ILVS to create a system called extILVS that scores more than pairwise relationships.[1)]

Local ILVS Method (LIM). Let L_i^{XY} denote a set of local structures that includes E_i^{XY} and let $L^{XY} = \cup_i L_i^{XY}$. For example, in Figure 3 let X=Gal1 and Y=Gal2. Then L_1^{XY} could be the causal structure shown in Figure 3. LIM (Local ILVS method) calculates $P(E_i^{XY} \mid D, K)$ by first, searching for the best L_i^{XY} that fits the data; and second, using all unique L_i^{XY} that were visited so far. Scores of the node pairs, calculated by extILVS, are used to guide the search for the best L_i^{XY}. Finally, we estimate Equation 2 by the following equation:

$$P(E_i^{XY} \mid D,K) \approx \frac{\sum_{S:E_i^{XY} \text{ is in } T} P(D,S \mid K)}{\sum_{T} P(D,T \mid K)} \qquad (6)$$

where T denotes all the structures generated in the search. Many heuristic methods have been used to search for the best structure that fits the data(Heckerman, Geiger et al. 1995). Note that unlike the previous methods, we concentrate on L_i^{XY}, i.e., the local structure of E_i^{XY}. In this paper we use structure search as defined in the following steps: (Step 1) Construct a set V that represents strongly related variables with X and Y. Let W equal $V \cup \{X,Y\}$. We limit the number of variables in W to be less than k and use those variables to define the structures in L^{XY}. Now any structure $S \in L^{XY}$ can be denoted as $S = \{E_i^P \mid P \in \{\text{all pairs in } W\}\}$. We initialize S to a random structure by randomly choosing E_i^P for all P. (Step 2) For a given structure S, we score with extILVS six different structures by substituting E_i^P with one of the six hypotheses (from Figure 2) for all node pairs P in W; (Step 3) Select the E_j^{P*} that in Step 2 generated the structure with the highest score; update S by substituting E_j^{P*} for E_i^P in S and repeat Step 2 with the new S. Stop the search if either there is no improvement in the structure score or the number of iterations exceeds a user defined limit. We also perturb the structure once the search reaches a local maximum. We provide three different perturbations to avoid a local maximum and they select one of the six hypotheses (from Figure 2) for each of the node pairs (X, Y) according to the following distributions:

- Random Perturbation: $\{P(E_i^{XY}|D,K) = 1/6 \mid i=1,2,\ldots 6\}$
- Local Perturbation: $\{P(E_i^{XY}|D,K) \mid i=1,2,\ldots 6\}$ calculated by LIM.
- ILVS Perturbation: $\{P(E_i^{XY}|D,K) \mid i=1,2,\ldots 6\}$ calculated by ILVS.

We later pair these perturbations to introduce different variations of LIM. Also, two different search methods were implemented:

- Local Search: Iterating Step 2 through Step 3 while forcing S to include pairwise relationships E_1^{XY}, E_2^{XY},…, E_6^{XY} for each of node pairs (X, Y)
- Global Search: Iterating Step 2 through Step 3 with no restrictions on S.

Example Run of LIM. For example, let us assume there are only five modeled nodes: U, V, X, Y, and Z. Further assume we are limiting $k = |W| < 4$ and $W=\{X, Y, Z\}$. In Step 1 we randomly initiate a structure, e.g., $S=\{E_1^{XY}, E_2^{XZ}, E_6^{YZ}\}$. In Step 2, we first consider the six different structures derived from S by substituting E_1^{XY} with any of $\{E_1^{XY}, E_2^{XY}, E_3^{XY}, E_4^{XY}, E_5^{XY}, E_6^{XY}\}$. We do the same for E_2^{XZ} and E_6^{YZ}. We evaluate all 18 different structures. In Step 3 we choose the highest scored structure and go to Step 2. Upon reaching a stopping condition, to score $P(E_1^{XY}|S,K)$, for example, we sum all scores of visited structures that include E_1^{XY} and divided by the sum of the scores of all visited structures. Note that indirect causal relationships, e.g., $X \leftarrow Z \leftarrow Y$, are also used in scoring E_1^{XY}.

Equivalence Local ILVS Method (EquLIM). In the ILVS model, the original set of three pairwise relationships between nodes X and Y allowed in a causal Bayesian network ($X \leftarrow Y$, $X \rightarrow Y$, and X Y) is extended to a set of six. The first three in the set of these pairwise relationships—defined as E_1, E_2, and E_3, respectively—are exactly these three original pairwise relationships. The last three relationships in this set—E_4, E_5, and E_6, respectively—represent the same directions of causality between X and Y as before, but also assume that a latent variable is directly causing both X and Y. Having this latent variable directly causing both nodes in a pairwise relationship is known as direct confounding.

In an ILVS Bayesian structure S, a latent variable directly confounding X and Y in a pairwise relationship can indirectly confound nodes A and B in S if this latent variable is d-connected to both A and B. Thus, even if no direct confounder exists between A and B (i.e., the direct relationship between A and B is either E_1, E_2, or E_3), a direct confounder between nodes X and Y in S could indirectly confound A and B, meaning the indirect relationship between A and B is classified as confounded

(i.e., the indirect relationship between A and B is either E_4, E_5, or E_6).

When trying to predict the causal relationship between nodes X and Y in an ILVS Bayesian structure S where there is at least one other node, the correct classification is their indirect relationship, which must take into account indirect confounding. Therefore, in order to correctly classify causality between X and Y in S, an algorithm must be developed to detect indirect confounding. The algorithm we used is outlined below.

We define the set of all undirected paths P<S,X,Y> from node X to node Y in structure S as if S is viewed as a graph rather than a digraph (i.e., directionality of arcs is not taken into account). One brute force approach to detecting indirect confounding between nodes X and Y would simply construct the set of all undirected paths P<S,X,Y>, but model all latent variables explicitly, and then test whether any of these latent nodes is d-connected to both X and Y. Another brute force approach would be to construct a single superstructure by modeling all latent variables explicitly and then testing whether any of these latent nodes are d-connected to both X and Y.

The indirect confounder-checking algorithm presented here examines the set of all undirected paths P<S,X,Y> from X to Y in structure S until either indirect confounding is found, or all paths are exhausted (signaling no indirect confounding). For each path, the algorithm starts at node X and steps through the sequence of pairwise relationships between node pairs P_1, P_2,…, P_n along that path. At any given step i, the algorithm notes only whether any pairwise relationships E_1, E_2, E_4, E_5, or E_6 have been encountered between any node pairs preceding node pair P_i in that path, making a total of only five flags to keep track of while advancing steps.

At each step i along the path, the algorithm first examines the pairwise relationship between node pair P_i. If it is E_1, then if E_2, E_5, or E_6 has been encountered between any node pair preceding P_i, any confounder between a node pair before P_i in this path cannot be d-connected to node Y and any confounder between a node pair after P_i in this path cannot be d-connected to node X. Thus, if any of these three pairwise relationships have been encountered between any node pair preceding P_i, the algorithm stops and moves to the next path. Otherwise, it moves on to the next pairwise relationship P_{i+1}, but resets the flag for whether E_4 has been found between any node pairs preceding P_i, since the confounder in an E_4 relationship found between a node pair preceding P_i cannot be d-connected to node Y.

If the pairwise relationship between node pair P_i is E_2, no direct confounder between a node pair after P_i can be d-connected to node X. Thus, if no direct confounder has been encountered in a node pair preceding P_i, the algorithm stops and moves to the next path. Otherwise, it moves on to the next pairwise relationship.

If the pairwise relationship between node pair P_i is E_4, if E_2, E_5, or E_6 has been encountered in a node pair preceding P_i, any confounder between a node pair before P_i in this path cannot be d-connected to node Y and any confounder between a node pair after P_i in this path cannot be d-connected to node X. If none of these conditions apply, the algorithm moves to the next pairwise relationship. Otherwise, it stops and moves to the next path.

If the pairwise relationship between node pair P_i is E_5, no combination of values for the five flags precludes indirect confounding between X and Y, so the algorithm always moves on to the next pairwise relationship.

If the pairwise relationship between node pair P_i is E_6, because there is no causal arc, if there is to be indirect confounding between X and Y in this path, the confounder in this E_6 relationship must be the cause. Therefore, if there have been any E_2, E_5, or E_6 relationships between node pairs preceding P_i, the confounder in this E_6 relationship cannot be d-connected to node X, so the algorithm stops and moves to the next path. Otherwise, it moves on to the next pairwise relationship, but resets all of the other flags since none of the relationships found between node pairs before P_i can cause indirect confounding between X and Y in this path.

If the algorithm reaches node Y while stepping through the node pairs on the path, it is guaranteed that at least one of the confounders found between node pairs along the path is d-connected to both X and Y. If an E_6 relationship was encountered between node pair P_i along the path, then there was only one of them. There were no E_2 or E_5 relationships between node pairs before P_i along the path and no E_1 or E_4 relationships between node pairs after P_i along the path. Therefore, the latent confounder in this E_6 relationship is d-connected to both X and Y and there is indirect confounding between them.

If an E_6 relationship wasn't encountered between any node pairs along the path, but an E_4 relationship was encountered between node pair P_i, no E_2 or E_5 relationships were found between node pairs before P_i along the path and no E_1 relationships were found between node pairs after P_i along the path. Therefore, even if multiple E_4 relationships were found between node pairs along the path, the confounder in the last one found is d-connected to both X and Y and there is indirect confounding between them.

If E_4 and E_6 relationships weren't encountered between any node pairs along the path, but an E_5 relationship was encountered between node pair P_i, there were only E_1 relationships (and/or E_4 relationships considered to be E_1) between node pairs before P_i and only E_2 and/or E_5 relationships between node pairs after P_i. Thus, in this case, the confounder in the first E_5 relationship encountered is d-connected to X and Y and there is indirect confounding between them.

However, it is possible that there were no confounders between any node pairs along the path. Therefore, when the algorithm reaches the end of the path, if E_4, E_5, or E_6 have been encountered between any node pairs along the path, there is indirect confounding between X and Y and the algorithm returns TRUE. If, however, no confounders were encountered between any node pairs along the path, the algorithm moves to the next path. If the algorithm exhausts the list of paths and fails to find indirect confounding, it returns FALSE.

4 EXPERIMENTAL METHODS

In evaluating causal learning, we ideally would know the real-world causal relationships (both the structure and parameters) among a set of variables of interest. With such knowledge we could generate experimental and observational data. Using these datasets as input, a learning method could predict the causal structure and estimate the causal parameters that exist among the modeled variables. These predictions and estimates would then be compared to the true causal relationships. Since confident knowledge of underlying causal processes is relatively rare, in this study of causal discovery from mixed high throughput data we used as a gold standard a causal model that was constructed by an expert biologist. In particular, we used gene regulation pathways in yeast *SNF*1 protein kinase (Schmidt and McCartney 2000), which was built based on a simulator that was built for producing high throughput. The next section describes how we generated data from the yeast *SNF*1 protein kinase simulator and then used this data in evaluating the learning method described in Section 3.

$$eX^t = eX^{t-1} + rate[-eX^{t-1} + F_X(\text{causes\_of}(X^t) \backslash X^{t-1})] + \varepsilon_X \quad (7)$$

4.1 DATA GENERATION

Only a few gene expression simulation systems are currently available (Tomita, Hashimoto et al. 1999; Saavedra and Glymour 2001; Scheines and Ramsey 2001). Limited functions are available in most of the systems because they are in their early development stages. For example, Tomita et al. (1999) 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) exporting the gene expression levels to a file, and (3) modeling of measurement errors.

We used the Scheines and Ramsey (2001) 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* (Edwards and Glass 2000) to update an expression level of a gene X:
where X^t represents the gene X at time t and eX^t represents the gene expression level of the gene X at time t, $0 < rate \leq 1$, causes_of(X^t) are the direct causes of X^t in the model, "\" is the set difference operator, ε_X is an error term drawn from a given probability distribution, and F_X is a binary function specified by the user (Edwards and Glass 2000). Binary functions have been used to model natural phenomena including gene causal pathway (Kauffman 1993). Also note that the model used in this evaluation study contain only a one-stage time-lag, an example of this is shown in Figure 5, i.e., if a gene has a causal relationship with another gene, it means the relationship is modeled as in Figure 5.

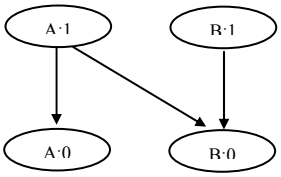

Figure 5. A one-stage time-lag model. A:0 represents the expression level of gene A at current time and A:1 represents the expression level of gene A at one time-step before the current time.

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 80 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 80 time lags for a burn-in period for evaluation study reported here.

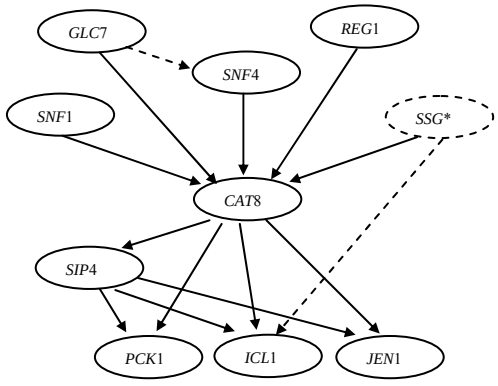

Figure 6. *SNF*1 simulation pathway model. Dotted lines represent the causal relationships that are biologically plausible, but need further investigation. *SSG represents a group of genes, i.e., *SIP*1, *SIP*2, and *GAL*83. *SSG* was modeled in the simulator but was hidden to the participants in the control study; i.e., the expression level of *SSG* was not provided to the participants**

We created a simulator model (Figure 6) using the SR simulator that models a gene regulation pathway based on assessments from a molecular biologist at our university who has many years of research experience related to gene regulation pathways in yeast *SNF*1 protein kinase (Schmidt and McCartney 2000). Regulation relationships

(e.g, *CAT*8 promotes *SIP*4) and other parameters of the SR simulator were assessed from the biologist. We estimated the measurement error from published yeast microarray data (Gasch, Spellman et al. 2000). GEEVE currently models gene expression level using discrete variables only, although it could be extended to model with continuous variables as well. Thus, we discretized each gene's expression level into three states (i.e., low, no change, and high) based on each gene's expression level (Yoo and Cooper 2002).

We have generated three datasets. Each dataset consists of 10, 50, or, 100 cases that were generated with no manipulation of genes, i.e., only observational data, using simulator shown in Figure 6. We have implemented EquLIM and LIM with C++ and used a linux machine to run EquLIM and LIM. Since EquLIM and LIM are anytime algorithms, we have let EquLIM and LIM run for about 3 hours for each dataset, resulting in the total computing time of 18 hours for the experiment.

5 EXPERIMENTAL RESULTS AND DISCUSSION

Here we show the AUROC results of LIM and EquLIM in analyzing the three datasets. To calculate the AUROC, we have calculated the AUROC under two prediction categories: *causal*, i.e., $E_1(E_2)$, and *independence*, i.e., E_3, predictions. Since we know the true pairwise causal relationships between all nine genes from Figure 6, we used all 36 pairs of genes to calculate whether LIM and EquLIM correctly predicted each relationship of the 36 pairs of genes.

We also show the causal ROC for the dataset with 10 cases because we believe most of the initial high throughput data studies (1) will have small number of cases (<20); and (2) will seek for novel *causal* relationships. Although in this paper we concentrate on small number of cases, we note that it was shown LIM correctly predicted pairwise causal relationships with a large dataset on ALARM Bayesian network (Cooper and Yoo 1999).

Table 1. The AUROC of LIM and EquLIM on three datasets, i.e., dataset with only observational data (10, 50, and, 100 cases). The results are shown for Local Structure Size of 5.

Algorithm	Cases / Prediction Type	10	50	100
LIM	Causal	0.4489	0.3640	0.3286
	Independence	0.2701	0.3571	0.4464
EquLIM	Causal	0.5812	0.4240	0.3350
	Independence	0.2254	0.3638	0.4561

Table 1 shows that the EquLIM predicts causal relationships better than LIM. It also shows that EquLIM predicts causal relationships significantly better than LIM if given a small number of cases, i.e., 10 cases. Although EquLIM does not predict independence relationships as well as LIM with 10 cases, given datasets with larger cases, EquLIM performs as well as LIM.

Figure 7 shows the causal prediction of ROC of EquLIM and LIM. It shows that EquLIM outperforms LIM in causal prediction.

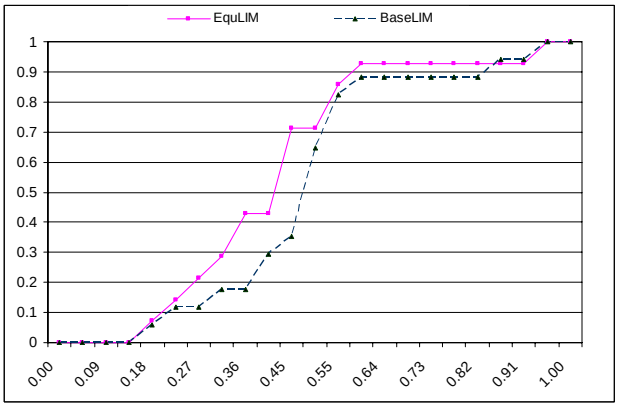

Figure 7. Causal prediction ROC for 10 observational cases

Although EquLIM showed promising performance compared to LIM, EquLIM requires more computing time than LIM. This additional time is allocated in searching for equivalence local structures. Current implementation of the equivalence local structures search is using recursive calls of subroutines. We note that non-recursive version of the search will further improve the performance of EquLIM.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have described a new Bayesian networks structure search algorithm called EquLIM. We have generated simulation data that includes no perturbation of a gene network. Better causal prediction abilities can be achieved by perturbations of the gene network. However, with limited budget and time, it is also useful to have an analysis that can predict novel causal relationships from limited resources. We believe EquLIM can be useful in such initial analysis of experiments with limited resources.

We have also shown that dataset with small cases (< 100) of only observational data is better analyzed – especially causal relationship predictions – with EquLIM than LIM that uses a greedy hill climbing Bayesian network structure search.

Extensions of this work include examining the synergy of case control data in conjunction with observational and experimental data (Cooper 2000), and modeling beyond pairwise causal relationships between the measured variables (Yoo and Cooper 2002). We also plan to apply EquLIM to actual experimental datasets.

Acknowledgments

The research was supported by NIH grant P20RR017670 from NCRR.

References

Cooper, G. F. (2000). A Bayesian method for causal modeling and discovery under selection. Proceedings of the Confernce on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Franciso, CA.

Cooper, G. F. and E. Herskovits (1992). "A Bayesian method for the induction of probabilistic networks from data." Machine Learning **9**: 309-347.

Cooper, G. F. and C. Yoo (1999). Causal discovery from a mixture of experimental and observational data. Proceedings of the Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann.

Edwards, R. and L. Glass (2000). "Combinatorial explosion in model gene networks." Chaos **10**: 691-704.

Gasch, A., P. Spellman, et al. (2000). "Genomic expression programs in the response of yeast cells to environmental changes." Mol Biol Cell **11**(12): 4241-57.

Heckerman, D. (1995). A Bayesian approach to learning causal networks. Proceedings of the Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann.

Heckerman, D., D. Geiger, et al. (1995). "Learning Bayesian networks: The combination of knowledge and statistical data." Machine Learning **20**: 197-243.

Kauffman, S. (1993). Origins of Order - Self-Organization and Selection in Evolution, Oxford University Press.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. San Mateo, CA, Morgan Kaufmann.

Saavedra, R. and C. Glymour (2001). A Regulatory Network Simulator. Simulator based on (Yuh et al. 1998) under development.

Scheines, R. and J. Ramsey (2001). Gene simulator. Available at: http://www.phil.cmu.edu/projects/tetrad/tetrad4.html.

Schmidt, M. and R. McCartney (2000). "beta-subunits of Snf1 kinase are required for kinase function and substrate definition." Embo Journal **19**(18): 4936-43.

Spirtes, P., C. Glymour, et al. (2000). Causation, prediction, and search. Cambridge, MA, MIT Press.

Tomita, M., K. Hashimoto, et al. (1999). "E-CELL: Software environment for whole cell simulation." Bioinformatics **15**(1): 72-84.

Yoo, C. and G. Cooper (2001). Causal discovery of latent-variable models from a mixture of experimental and observational data. Center for Biomedical Informatics Research Report CBMI-173. Pittsburgh, PA, Center for Biomedical Informatics.

Yoo, C. and G. Cooper (2002). Discovery of gene-regulation pathways using local causal search. AMIA, San Antonio, Texas.

Yoo, C., V. Thorsson, et al. (2002). Discovery of a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. Pacific Symposium on Biocomputing, Maui, Hawaii, World Scientific.