Exploring gene causal interactions using an enhanced constraint-based method

Xintao Wu*, Yong Ye

*Corresponding author. Tel.: +1 704 687 8586; fax: +1 704 687 3516.
E-mail addresses: xwu@uncc.edu (X. Wu), yye@uncc.edu (Y. Ye).

1. Introduction

Whole-genome expression profiling, facilitated by the development of DNA microarrays [1], represents a major advance in genome-wide functional analysis. The prediction and identification of physical and genetic interactions from gene expression data are some of the most challenging tasks of modern functional genomics. Gene interaction networks contain information about which pathways a gene belongs to and which genes it interacts with. Moreover, gene interaction information can be used to predict the functions of a gene in terms of how it interacts with other genes as well as to indicate the pathway initiators which are potential drug targets.

Although various interaction analysis methods (e.g., causal probabilistic networks [2,3], graphical Gaussian modeling [4], loglinear modeling [5], association rule [6] etc.) have been well studied in data mining and statistics, we face new challenges in applying these methods to the analysis of microarray data. First, genes in rows of gene expression matrices\(^1\) are of very high dimensionality (e.g., \(10^3\)–\(10^4\) genes) while samples in columns are of relatively low dimensionality (e.g., \(10^1\)–\(10^2\) samples). Thus, microarray data sets are very sparse in high-dimensional gene spaces. Second, the microarray data are usually corrupted with a substantial amount of measurement noise. Third, the nature of genetic networks is undoubtedly stochastic but microarray measurements are population averaged, which mask the real individual regulatory interactions [7].

These three problems make it extremely unreliable to have a straightforward estimation of model parameters. It is obvious that if the complexity of the model is increased sufficiently, the model will describe the underlying

\(^1\) Here the raw microarray images are transformed into gene expression matrices, where rows denote genes and columns denote various samples, conditions, or time points.
complex biological processes more accurately. Clearly, when only a fixed amount of measurements is available, more parameters will generally mean less reliable estimates. Therefore, there is a great need for new tools to explore and analyze complex gene interactions in a highly interactive environment, which requires the computational aspects of modeling to be both efficient and effective.

Bayesian network has recently been investigated for gene regulatory networks [8–10]. The causal network learnt can present the knowledge embedded in the microarray data in a manner that is intuitive and familiar to biologists. However, the difficulty with this technique is that learning the Bayesian network structure is an NP-hard problem, as the number of directed acyclic graphs (DAGs) is superexponential in the number of genes, and an exhaustive search is intractable. In Ref. [11], Peter Spirtes and Clark Glymour proposed a constraint-based method, \textit{PC} (Peter–Clark), which starts from the complete undirected graph, then thins that graph by removing edges according to conditional independence relations in various orders. However, PC needs a long time to finish when a large number of variables are present, which makes interactive exploration of microarray data impossible.

In this paper, we investigate an enhanced constraint-based approach to learn causal interactions from microarray data. As constraint-based causal learning approaches are with high computational cost, we use the undirected independence graph which is the output of graphical Gaussian modeling, rather than the complete graph suggested in PC. As it becomes infeasible to use graphical Gaussian modeling when the number of genes exceeds that of samples, we apply existing clustering methods or association rule mining prior to using graphical Gaussian modeling. To further improve performance and support better interactive exploration, we apply graph decomposition techniques to decompose a large independence graph into subgraphs and apply causal modeling method on each component to derive causal interactions, respectively.

The remainder of the paper is structured as follows. In Section 2 we review related work. Section 3 revisits constraint-based causal modeling method. Section 4 presents our enhanced constraint-based causal learning method. Experimental results are discussed in Section 5. In Section 6 we draw conclusions and describe directions for future work.

2. Related work

Unsupervised learning procedures such as clustering has been widely and successfully used to identify families of genes so far. Various traditional clustering algorithms have been applied, and some new methods [12–15] were proposed to find co-expressed genes from microarray data. The above work models a gene expression data set as a weighted graph where each gene is represented by a vertex and each edge denotes the similarity between two genes. The classical graph-theoretical algorithms (e.g., the minimum cut, the maximum clique, or the minimum spanning tree, etc.) are applied to this weighted graph.

Ben-Dor et al. [12] propose a corrupted clique graph data model. The input data set is assumed to come from the underlying cluster structure with random errors. A heuristic algorithm \textit{CAST} (cluster affinity search techniques) is proposed to iteratively identify the corrupted cliques one at a time. Hartuv et al. [13] develop a method called \textit{HCS} (highly connected subgraph). It recursively splits the weighted graph into a set of highly connected components based on the minimum cut. Each highly connected component is then considered as a cluster. Similarly, Shamir et al. [14] develop a method called \textit{CLICK} (cluster identification via connectivity kernels), which includes how to measure the coherence within a subset of genes and determine the criterion to stop the recursive splitting process. Xu et al. [15] develop a method called \textit{MST} (minimum spanning tree). It first generates a minimum spanning tree from the weighted graph and then removes $k-1$ edges from the generated spanning tree. The isolated $k$ subtrees are considered as $k$ clusters.

Gene clusters from various clustering methods can be interpreted as a network of co-regulated genes, which may encode interacting proteins that are involved in the same biological processes. However, clustering methods cannot address at least two issues: (1) they do not indicate gene interactions and (2) they cannot identify molecular networks or analyze high-level function, i.e., the gene expression changes in the context of biological pathways; this is because the clustering methods cannot give the information on how one gene influences other genes in the same cluster.

In Ref. [16], association rules [17] are applied to investigate how the expression of one gene may be associated with the expression of a set of genes. The kind of rule that can be discovered is, for example, “when gene A and gene B are over-expressed within a sample, then often gene C is also over-expressed”. However, the association rule method can only capture gene co-expression, and not interactions because it is exclusively based on support measure. Recently, log-linear modeling has been investigated to overcome the limitations of support-based association algorithms [18] and was applied in microarray data analysis [19]. This method can be used to reveal all significant high-order non-linear combinatorial interactions (e.g., a gene is over-expressed only if several genes are jointly over-expressed, but not if at least one of them is not over-expressed). However, both association rule and log-linear modeling require a discretization of the data, which may cause some information loss during discretization. Furthermore, interactions learnt by these two methods are undirected instead of the directed causal interactions investigated in this paper.

Log-linear modeling can be used to derive non-linear combinatorial interactions (undirected) and we presented results
3. Constraint-based causal modeling revisited

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$ be a set of samples or conditions and $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$ be a set of genes. The microarray data can be represented as $\mathcal{X} = \{x_{ij}\}_{i=1}^n j=1, \ldots, m\} (n \gg m)$, where $x_{ij}$ corresponds to the expression value of gene $g_i$ in sample $s_j$. In this section, we revisit constraint-based causal modeling which is used in our paper.

The constraint-based approaches use data to make categorical decisions about whether particular conditional-independence constraints hold and then pieces these decisions together by looking for those sets of structures that are consistent with the constraints. In Ref. [2] Spirtes et al. proposed a $PC$ constraint-based method which starts from the complete undirected graph, then thins that graph by removing edges with zero order conditional independence relations, thins again with first order conditional independence relations, and so on. The set of variables conditioned on need only to be a subset of the set of variables adjacent to one or the other of the variables conditioned. Fig. 1 shows the algorithm (the correctness of the procedure is given by Theorem 3.4 on p. 47 of Ref. [2]). The procedure to orient undirected edge is shown from lines 13 to 20. Please note that $d$-separation plays an important role in constraint-based algorithm. For any three disjoint vertex sets $X$, $Y$ and $W$ in a directed graph, $X$ and $Y$ are $d$-separated given $W$ if there is no active adjacency path between $X$ and $Y$. An adjacency path is a path between two vertices without considering the directionality of the arcs. A path between $X$ and $Y$ is active given $W$ if: (1) every collider in the path is in $W$ or has a descendant in $W$ and (2) every other vertex in the path is not in $W$. A collider of a path is a node where two arcs in the path meet at their endpoints.

The complexity of the $PC$ constraint-based causal modeling method is bounded by the largest degree in the undirected graph. In the worst case, the number of conditional independence tests required by the algorithm is bounded by $(n^2(n - 1)^{k-1})/(k - 1)!$ where $k$ be the maximal degree of any vertex and $n$ be the number of vertices [2].

4. Our enhanced $GGM + PC$ method

The goal here is to explore interactions between genes interactively and learn genetic networks from microarray data. Our method is outlined as follows:

1. $Preprocessing$: We subject the input data $\mathcal{X}$ to clustering or association rule mining, prior to analyzing gene interactions.
2. $GGM$: For each cluster
   - Compute the variance matrix $\mathcal{V}$ where $v_{ij}, i, j = 1, \ldots, n$, corresponds to covariance between gene $g_i$ and $g_j$.
   - Compute its inverse $\mathcal{S} = \mathcal{V}^{-1}$.
   - Scale $\mathcal{G}$ to have a unit diagonal and compute partial correlations $pr_{x_i x_j g}$. 
   - Draw the independence graph according to the rule that no edge is included in the graph if the absolute value of the partial correlation coefficient is less than some threshold $s$.
3. $Decomposition$: For those graphs with large numbers of vertices, decompose those independence graphs into smaller components.
4. $Causal modeling$: For each component, apply $PC$ (lines 7–20 of Fig. 1) with its independence graph as input.

We have two major enhancements. First, we use the undirected independence graph which is the output of graphical Gaussian modeling, rather than the complete graph suggested in the $PC$ algorithm, as input. The advantage of graphical Gaussian modeling is that it can generate an undirected independence graph for a relatively large set of genes very quickly. The independence graph is much simpler than the complete graph, which can significantly decrease searching complexity. Second, the independence graph can be further decomposed into components and the causal analysis is done over each small component. We leave the discussion of this part to Section 4.2. The motivation here is that though many biological pathways and processes are known to involve interactions among a relatively large number of genes, the genetic network topology is dominated by a few highly connected nodes which link the rest of the less connected nodes.
One potential problem is that it becomes infeasible to use graphical Gaussian modeling when the number of genes exceeds that of samples. This is because the correlation matrix is generally degenerate as the matrix rank is bounded by the sample size. In most microarray data sets, we usually have more than thousands of genes but hundreds of samples at most. In our system, we can apply various existing clustering methods and frequent item set mining to get gene clusters before applying graphical Gaussian modeling. We will discuss how this preprocessing affects interactive gene interaction analysis in Section 5. The number of genes contained in the resulting clusters or frequent item sets is expected to be less than the size of samples, thus avoiding the matrix rank problem. The authors, in Refs. [21,22], propose multiple regression procedures with variable selection to get approximate partial correlations between any pair of genes. However, multiple regression procedures are infeasible for microarray data sets with thousands of genes because of high computational cost.

4.1. Graphical Gaussian modeling

Graphical Gaussian model [4,23], also known as covariance selection model, assumes a multivariate normal distribution for the underlying data and satisfies the pairwise conditional independence restrictions which are shown in the independence graph of a jointly normal set of random variables. The graphical Gaussian modeling is feasible for microarray data analysis because the log transformed microarray expression data nearly satisfy multivariate normality due to the nature of experimental errors. Although there is no dearth of literature disputing the most
appropriate distribution of gene expressions, many researchers take natural or base-2 log transforms of the raw expression data to impose normality on expression levels [22].

The independence graph is defined by a set of pairwise conditional independence relationships that determine the edge set of the graph. A crucial concept of applying graphical Gaussian model is partial correlation, which measures the correlation between two variables after the common effects of all other variables in the genome are removed.

\[ pr_{xyz} = \frac{r_{xy} - r_{xz}r_{yz}}{\sqrt{(1 - r_{xz}^2)(1 - r_{yz}^2)}}. \]  

Eq. (1) shows the form for the partial correlation of two genes, \( g_x \) and \( g_y \), while controlling for a third gene variable \( g_z \), where \( r_{xy} \) denotes Pearson’s correlation coefficient. The partial correlation, \( pr_{xy} \), of genes \( g_x \) and \( g_y \) with respect to gene \( g_z \) may be considered to be the correlation \( r_{xy} \) of \( g_x \) and \( g_y \), after the effect of \( g_z \) is removed. If there is no difference between \( pr_{xy} \) and \( r_{xy} \), we can infer that the control variable \( g_z \) has no effect. If the partial correlation approaches zero, the inference is that the original correlation is spurious (i.e., there is no direct causal link between the two original gene variables because the control gene variable is either the common anteceding cause, or the intervening variable).

Partial correlations that remain significantly different from zero may be taken as indicators of a possible causal link.

Given a set of genes \( g \), the partial correlation can be computed by \( pr_{xyg} = -s_{xy}/\sqrt{s_{xx}s_{yy}} \), where \( s_{xy} \) is the \( xy \)th element of the inverse of variance matrix (\( S = S^{-1} \)). It is known that conditional independence constraints are equivalent to specifying zeros in the inverse variance [23]. Under the assumption of multivariate normality, tests for vanishing partial correlation are equivalent to tests for conditional independence, i.e., \( pr_{xyg} = 0 \iff x \perp y | g \). In our paper, we apply Fisher’s \( z \) test,

\[ z(pr_{xyg}, n) = \frac{1}{2} \sqrt{n - |g| - 3} \times \ln \left( \frac{1 + pr_{xyg}}{1 - pr_{xyg}} \right), \]

where we use \( n \) to denote the number of samples and use \( |g| \) to denote the number of variables in the set \( g \). The independence graph is then determined according to the rule that no edge is included in the graph if Fisher’s \( z \) test is less than the specified threshold. It is important to note that partial correlation is different from standard correlation, indicates better evidence for genetic regulatory links than standard correlation, as evident from our previous results [19], and is in agreement with biological interpretation.

4.2. Graphical decomposition

As we presented in Section 4.1, graphical Gaussian models assume a family of normal distributions for the underlying data constrained to satisfy the pairwise conditional independence restrictions inherent in the independence graph. It has become increasingly clear that signaling pathways interact with one another and the final biological response is shaped by interaction between pathways. Genes functioning in the same biological process/pathway are usually co-expressed and tightly regulated, both spatially and temporally [24]. Gene interactions within one pathway are generally strong because they need to form one unit and perform a certain biological function while the inter-pathway interactions are much weaker since different pathways within distinct functions normally do not strongly interact with each other [25]. Hence, we expect that the relatively large independence graph generated from graphical Gaussian modeling may show the interactions among different pathways and the inter-pathway interactions are not as strong as gene interactions within each pathway. This suggests that we may decompose the independence graph into basic, irreducible components and each component may respond to a single pathway.

Graph-theoretical results show that if a graph corresponding to a graphical model is decomposable into subgraphs by a clique separator, the maximum likelihood estimates for the parameters of the model can easily be derived by combining the estimates of the models on the simpler subgraphs. Hence, applying a divide-and-conquer approach based on the decompositions will make the procedure applicable to much larger subsets of genes.

The theory may be interpreted in the following way as shown in Fig. 2: if two disjoint subsets of vertices \( S_a \) and \( S_b \) are separated by a subset \( S_c \) in the sense that all paths from \( S_a \) to \( S_b \) go through \( S_c \), then the variables in \( S_a \) are conditionally independent of those in \( S_b \) given the variables in \( S_c \). The graph may be further decomposed into subgraphs \( S_a \cup S_c \) and \( S_b \cup S_c \). The requirement that the subgraph on \( S_c \) be complete implies that there is no further independence.
constraints on the elements of $S_e$, so that this factorization contains all the information about the joint distribution. To find the clique separators of a graph or to find the vertex-sets of the irreducible components of the graphs, an algorithm with a complexity of $O(ne + n^3)$ [26] is implemented in our system, where $n$ is the number of vertices and $e$ is the number of edges. As discussed in Section 3, the complexity of PC algorithm in the worst case is bounded by $(n^2(n-1)^{k-1})/(k-1)!$, where $k$ be the maximal degree of any vertex and $n$ be the number of vertices. Our $PC+GGM$ method decreases complexity by applying graphical Gaussian modeling and graphical decomposition. As we know, the complexity of applying graphical Gaussian modeling to construct an independence graph is $O(n^3)$, where $n$ is the number of nodes, since the matrix inversion has $O(n^3)$ complexity. The combination of graphical Gaussian modeling and graphical decomposition can generate a set of subgraphs with much smaller $k$ (maximal degree of any vertex) and $n$ (the number of vertices) values.

5. Experimental results

The experiments were conducted in a DELL Precision 340 Workstation (Redhat Linux 9.0 operating system), with one 2.4 GHz processor, and 1 G bytes of RAM. We used two data sets in our experiments. The first one is Yeast data set which contains expression profiles for 6316 transcripts corresponding to 300 diverse mutations and chemical treatments in yeast [27]. The second is E. Coli data set, which includes 4289 genes and 52 samples [28].

In our system, we first apply some data mining tools to get subsets of genes. For each subset, the independence graph is generated by using graphical Gaussian modeling. The independence graph is then decomposed to get subgraphs and each subgraph is then used as the input for causal analysis. We use automatic graph drawing tools [29] to represent gene networks and our implementation is in C++ on Linux workstations using FLTK [30] for the user interface. During this process, the users may interactively add or remove some genes from the independence graph. A preliminary version of our prototype system is available via (http://www.cs.uncc.edu/~xwu/bio/GenExplore.html).

5.1. Preprocessing

Preprocessing is one important and necessary step to get subsets of genes for further interaction analysis since the number of samples is usually much less than that of genes. Generally, any clustering or frequent itemset mining method can be used here. In our system, we have integrated two public available data mining softwares in our system. The first one is frequent itemset mining from Ref. [31] while the second one is hierarchical clustering from Ref. [32].

To apply frequent itemset mining, we consider each gene expression sample as a single transaction and each transcript or gene as an item. Since in an expression sample each transcript or gene is assigned a real value that specifies the relative abundance of that transcript or gene in the profiled sample, we need to discretize each measured real value as being up (i.e., over-expressed), down (under-expressed), or neither up nor down. In this case, any particular gene in a sample can be thought of as being two items, one item referring to the gene being over-expressed, the other referring to the gene being under-expressed. A gene expression sample (transaction) would include the set of genes that are either over-expressed or under-expressed in the sample. A commonly used algorithm for frequent itemset mining is the a priori [17] which applies a priori property, i.e., every subset of a frequent itemset must also be a frequent itemset. The algorithm proceeds iteratively, first identifying one-item frequent itemsets, generating two-item candidate frequent itemsets by extending frequent itemsets identified in the previous step with one more item, and applying one single scan of the data set to determine which candidates are frequent itemsets.

In Ref. [16], the yeast data set is transformed by binning an expression value greater than 0.2 for the log base 10 of the fold change as being over-expressed; a value less than $-0.2$, as being under-expressed; and a value between $-0.2$ and $0.2$ as being neither over-expressed nor under-expressed. We apply the same discretization strategy in our experiment. E. Coli data set is transformed by binning an expression value greater than 1.2 as being over-expressed; a value less than $-1.2$ as being under-expressed; and a value between $-1.2$ and $1.2$ as being neither over-expressed nor under-expressed. Table 1 shows the size of gene sets obtained using frequent itemset mining method on both Yeast and E. Coli data sets with different support values. We can see the size of frequent itemset under low support values is large. Table 1 also shows the execution time of frequent itemset mining. We can see the execution time of preprocessing is generally trivial compared with interactive causal interaction analysis.

We also apply the hierarchical clustering method on these two data sets. The goal of this algorithm is to compute a dendrogram that assembles all elements into a single tree. An upper-diagonal similarity matrix which contains similarity

<table>
<thead>
<tr>
<th>Data set</th>
<th>Support (%)</th>
<th>No. frequent itemsets</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>12</td>
<td>1635</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>795</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>298</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>164</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>69</td>
<td>0.10</td>
</tr>
<tr>
<td>E. Coli</td>
<td>18</td>
<td>182</td>
<td>254</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>53</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>12</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>2</td>
<td>0.01</td>
</tr>
</tbody>
</table>
scores for all pairs of genes is computed first. The matrix is then scanned to identify the highest value (representing the most similar pair of genes). A node is created joining these two genes, and a gene expression profile is computed for the node by averaging observation for the joined elements. The similarity matrix is updated with this new node replacing the two joined elements, and the process is repeated.

The execution times of hierarchical clustering on Yeast and E. Coli are 7205 and 4500 s, respectively.

One natural question is how to evaluate the preprocessing results obtained from frequent itemset mining or hierarchical clustering methods since the results will significantly affect the following interaction analysis. Although many clustering methods have been proposed for microarray, to our best knowledge, there is no literature systematically evaluating how clustering results match known complexes (pathways) identified by biologists. In our experiment, we compared our preprocessing results with previously known complexes of Yeast and E. Coli. 451 known Yeast complexes, which cover 1230 genes, are presented in Ref. [33] while 530 know E. Coli complexes, which cover 1146 genes, are presented in Ref. [34]. Our results show that most frequent itemsets or clusters do not match well with known complexes published in Refs. [33,34]. For example, there are only five out of 182 E. Coli frequent itemsets (as shown in Table 1 using support value 18%), which match some known complex. The reason has twofold. First, the known complexes of both Yeast and E. Coli only cover a small fraction of genes (e.g., 19.5% for Yeast and 23.7% for E. Coli). Second, the results using frequent itemset mining or hierarchical clustering may not be accurate due to the small size of samples and the high noise contained in microarray data. Hence some co-related genes may not be clustered into the same clusters. This suggests interactive interaction analysis tools which can incorporate a priori knowledge to support explorative operations (e.g., adding or removing genes from a cluster) are greatly needed to discover unknown complexes (pathways).

5.2. Performance analysis

Fig. 3 shows the execution time of our enhanced method of PC + GGM for gene clusters with different sizes. We randomly chose five clusters with size 30, 70, 100, 150, and 260, respectively. Please note that the maximum of genes contained in one single cluster is 300 as we only have 300 samples in this Yeast data set. For each cluster, we varied thresholds used for vanishing partial correlation in the independence graph and reported their execution times, respectively. We also included the execution time of the original PC method. As the original PC method uses complete graph as input and does not use graphical decomposition, its effect is equivalent to our GGM + PC with threshold s = 0. Fig. 3 shows that for all Yeast clusters with size varying from 30 to 260, the result of our enhanced method GGM + PC is significantly better than that of the original PC. For example, our GGM + PC needs few seconds to process a cluster with 150 genes while the original PC needs 220 s. This significant improvement of performance makes interactive exploration feasible. For a cluster with 260 genes, the original PC needs more than 5 h while our enhanced method with s = 0.2 needs only about 1 min. This is because (1) our GGM + PC removes those edges which could not be causal interactions from the independence graph directly and (2) our graphical decomposition decreases the complexity.

5.3. Causal interactions

5.3.1. Comparison with PC method

Table 2 shows the comparison between graphs generated using our PC + GGM method and graphs generated using the original PC method. To compare the similarity of two graphs, we use the traditional measures: sensitivity and specificity. The sensitivity, se, is the ratio of correctly identified edges over the total number of edges in the original network. The specificity, sp, is the ratio of edges correctly identified as not belonging in the graph over the true number of edges not present in the original network. An algorithm can achieve perfect sensitivity or specificity by including or excluding, respectively, all edges from the output. In this paper, we apply a combined statistics, \( d = \sqrt{(1 - se)^2 + (1 - sp)^2} \) proposed in Ref. [35], to measure the similarity between two graphs.

In this experiment, we randomly chose four components with sizes 10, 20, 47, and 84, respectively. For each component, we ran our PC + GGM method with varying threshold s and compared the generated graph with the graph generated by the original PC using the measures described as above. We can observe, for all components, the smaller the threshold s we set, the more similar between the generated
graphs. For example, when we set threshold $s$ as 0.01, the distances are less than 0.15 under all four cases, which suggested the generated graphs are very similar. This is because more undirected edges are left in the independence graph generated by graphical Gaussian modeling. However, when we increase $s$ to 0.2 for the first two components (with 10, 20 genes, respectively) or increase $s$ to 0.1 for the last two components (with 47, 84 genes, respectively), the generated graph is generally quite different from that generated by using the original PC method. How to determine an optimal threshold $s$ for a given component remains as our future work.

5.3.2. Comparison with protein interaction databases

The performance of our method was further evaluated by how well our predictions agree with the currently available protein interaction databases. As an example, we chose the cluster that shows potential interactions between SNO and SNZ family proteins. Fig. 4(a) shows independence graph generated from graphical Gaussian modeling for one selected gene group with 12 genes (we omit biological information for each gene due to space limitation). Briefly, nine genes have known functions and seven genes encode proteins that are involved in biosynthesis/metabolism. Note we used dashed lines to indicate a negative partial correlation and solid line to indicate a positive partial correlation. The negative correlation (e.g., between YMR095C and YJR025C) of the biosynthesis/metabolism pathways or their pair of genes may counteract with each other (activators and repressors) of the biosynthesis/metabolism pathways or their expression is negatively regulated by the other gene in each pair.

5.3.3. Comparison with Bayesian-based approaches

We first run our constraint-based method on ALARM data, a Bayesian network used in medical diagnosis decision support system [37]. Ten thousand training instances were generated by randomly sampling from the distribution of ALARM. The original ALARM graph contains 37 variables and 46 edges. Our generated graph contains 56 edges. We achieve 0.89 and 0.98 in our generated graph for sensitivity and specificity, respectively. Thus the distance between our generated graph with original graph is 0.11, which suggests our constraint-based approach can effectively discover causal structures.

However, we got very different causal networks on yeast data using constraint based and Bayesian-based approaches. Most existing structure learning tools based on Bayesian approach are not feasible here due to a relative large

---

\*A list of tools is available at [http://www.ai.mit.edu/~murphyk/Bayes/bnsoft.html](http://www.ai.mit.edu/~murphyk/Bayes/bnsoft.html).
number of variables. We chose LibB [38] in our experiment. **Fig. 5** shows causal network learnt from LibB. Clearly it is very different from that of PC approaches as shown in **Fig. 4.** Although we cannot conclude constraint-based approaches are better than Bayesian-based approaches because there are currently no publicly available genetic regulatory networks used for benchmarks. One problem with causal learning using LibB is information loss due to discretization. The causal networks learnt may vary significantly when different discretization strategies are applied. Furthermore, it used a fixed ordering of the variables to reduce the computational cost, which makes results uncertain.

6. Conclusions and future work

In this paper we presented an enhanced causal structure learning approach, which allows users to analyze
complex gene interactions in a highly interactive environment, permitting exploration. Experiments demonstrated that our method is sensitive enough to detect relatively less well-documented protein–protein interactions and is able to detect protein–protein interactions that can be validated using existing protein interaction databases. Therefore, we expect that the interactions predicted by our methods that do not exist in current databases represent potentially new interactions that need to be verified using experimental approaches and will potentially lead to novel discoveries.

There are some aspects of this work that merit further research. Among them, we are trying to relieve the dimensionality problems by incorporating a priori knowledge during the modeling process. From the existing protein–protein interaction databases, users may already know some causal interaction relationships. Those a priori known relationships will be incorporated as the input, which can make the method more effective and efficient. Another research problem here is that, in constraint-based approaches, the output depends on the threshold used in independence tests. For causal conclusions to be correct asymptotically, the threshold must be adjusted as a function of sample size. We will also incorporate various clustering methods in our system. Finally, we will compare our constraint-based approach with Bayesian-based approach in microarray data where gene expressions usually are noisy and stochastic.

Acknowledgments

We gratefully thank anonymous reviewers and editor for their constructive comments. This work is an extension of Ref. [39] and was in part supported by funds provided by the University of North Carolina at Charlotte. We would like to thank Dr. Liying Zhang (Memorial Sloan Kettering Cancer Center) for her help on providing biological explanations on gene interactions illustrated in the paper.

References

[29] J. Elison, S. North, Graph visualization project (graphviz) [http://www.graphviz.org].


[38] N. Friedman, G. Elidan, Libb, a tool for learning bayesian network (*http://www.cs.huji.ac.il/labs/compbio/LibB*).


**About the Author**—XINTAO WU is an Assistant Professor of Computer Science Department at University of North Carolina at Charlotte. He got his Ph.D. in Information Technology from George Mason University in 2001. He received his BS degree in Information Science from the University of Science and Technology of China in 1994, an ME degree in Computer Engineering from the Chinese Academy of Space Technology in 1997. His major research interests include data mining and knowledge discovery, data privacy and security, and bioinformatics.

**About the Author**—YONG YE is a Ph.D. student of Computer Science Department at University of North Carolina at Charlotte. He got his BE in Computer Science from Beijing Polytechnic University in 1998. His major research interests include data mining and bioinformatics.