Modelling gene functional linkages using yeast microarray data

Tie Wang and Guoliang Xue
Department of Computer Science and Engineering, Arizona State University, Tempe, Arizona 85287, USA
E-mail: tie.wang@asu.edu
E-mail: xue@asu.edu

Jeffrey W. Touchman*
Center for Evolutionary Functional Genomics, The Biodesign Institute, Arizona State University, Tempe, Arizona, 85287-5301, USA
E-mail: jtouchman@tgen.org
*Corresponding author

Abstract: Understanding how genes are functionally related requires efficient algorithms to model networks from expression data. We report a heuristic search algorithm called Two-Level Simulated Annealing (TLSA) that is more likely to find the global optimal network structure compared to conventional simulated annealing and other searching schemes. We have applied this method to search for a global optimised network structure from a synthetic data set and an expression data set of *S. cerevisiae* mutants. We have achieved better precision and recall compared to other searching algorithms and are able to map relationships more accurately among functionally-linked genes.

Keywords: DNA microarrays; sources of variation; replication; correlation; differential expression analysis; ANOVA; bioinformatics.


Biographical notes: Tie Wang is a Graduate student pursuing a PhD Degree at Arizona State University in the Department of Computer Science. He received his BS in Electrical Engineering from the Harbin Institute of Technology, China in 1995 and MS in Computer Science from University of Arkansas in 2002. His research interests include combinatorial and machine learning algorithms and their applications in bioinformatics.

Guoliang Xue is a Professor in the Department of Computer Science and Engineering at Arizona State University. He received the PhD Degree in Computer Science from the University of Minnesota in 1991 and has held previous positions at the Army High Performance Computing Research Center and the University of Vermont. His research interests include efficient optimisation algorithms, with applications to networking and bioinformatics. He has published over 140 papers in these areas and has been continuously...
Modelling gene functional linkages using yeast microarray data

supported by federal agencies including NSF, ARO and DOE. He is the recipient of an NSF Research Initiation Award in 1994 and an NSF-ITR Award in 2003.

Jeffrey W. Touchman is an Investigator at the Translational Genomics Research Institute in Phoenix, Arizona, USA where he is Head of the Systems Biology Unit and Director of DNA Sequencing. He is also an Assistant Professor of Biology and Associate Director of the Center for Evolutionary Functional Genomics at Arizona State University. He obtained his PhD from the University of Texas and was a Postdoctoral fellow at the National Human Genome Research Institute at the National Institutes of Health (NIH) in Bethesda, Maryland, USA. His present research is focused on development of RNAi-based therapeutics and modelling of genetic regulatory networks. He is a recipient of the NIH Director’s Award in 1998.

1 Introduction

High-throughput molecular biology has motivated the development of algorithms and tools for analysing gene expression data, a central goal of which is to understand the interaction patterns between genes. Inference of genes that are functionally associated has been an active focus of research and several methods have been developed to predict or verify network associations, such as linear models (D’Haeseleer et al., 1999), Boolean networks (Akutsu et al., 1999; Ideker et al., 2002; Liang et al., 1998; Shmulevich et al., 2002), Bayesian Networks (BN) (Murphy and Mian, 1999; Hartemink et al., 2001, 2002; Friedman et al., 2000; Pe’er et al., 2001) and differential equations (Chen et al., 1999; Mestl et al., 1995). To varying degrees, these methods have been successful at learning genetic networks, yet the prospect of improving the accuracy of gene network models derived from real data still represents a present challenge.

Learned network structures from microarray data often do not reflect the correct causal relationships among genes unless ancillary ordering information is included in the analysis, such as prior biological knowledge of gene relationships. One significant reason for this is that there may be multiple graphs that have the same posterior score as the inferred one (Neapolitan, 2004). In this work, we focus on elucidating functional linkages among genes. The directed graphs in the inferred network reflects functional linkages rather than conditional dependencies among connected gene pairs. We present a principled method to discover the relationships among genes using expression data. In our modelling framework, the networks that map these gene relationships are represented as BN that can robustly capture the joint distribution of ontologically related factors. A Bayesian model combined with searching algorithms is used to infer linkages among the genes that are mapped onto networks based on their expression levels.

The problem of finding gene relationship structures from expression data can be viewed as a searching problem using heuristic algorithms. The structure that is associated with the optimised score value reflects the best network structure that we can learn from the data. Traditional simulated annealing is a heuristic search algorithm for finding a global minimiser of a function and it has been used to find genetic network structures (Ideker et al., 2002; Hartemink et al., 2001; Shou et al., 2003; Smith et al., 2003). Here, we apply an improved method, TLSA (see Section 2) (Xue, 1994), to infer gene
functional linkage graphs using microarray expression profiles. We originally applied TLSA to the study of molecular confirmation (Xue, 1994). TLSA performs perturbations at a high level and makes the decision at the local minimum (obtained by a local minimisation algorithm) starting from an initial point at the higher level. As such, it is much less likely to be entrapped into a local optimiser even at low temperatures. To the best of our knowledge, it is the first improved method based on simulated annealing applied to inferring gene functional linkages. We note that Ott et al. (2004) have reported a dynamic programming method to find the global optimum of a BN. However, the high time complexity limits its application to studying small sets of genes if no prior knowledge regarding their interactions is applied. The TLSA method does not require any prior knowledge and aims to find an undirected graph reflecting the optimal connections among genes. Thus it has a reduced search space and works well empirically on relatively larger sets of genes.

Given a microarray data set, the TLSA method can construct the structure of networks that map the putative functional relationships among genes, and the structure that is associated with the optimal Bayesian Dirichlet equivalence (BDe) score reveals the gene connections in the network. We have applied our method to a synthetic data set and data from *Saccharomyces cerevisiae* (Hughes et al., 2000). The results show high precision and recall values compared to other heuristic search methods. We also exploit Gene Ontology (GO) information to estimate the statistical significance of predicted graphs mapping functional linkages among genes.

2 Methods

2.1 Algorithm of TLSA

**Algorithm 1:** Two-Level Simulated Annealing ($G_{init}, T, \alpha$). It is assumed that there is an efficient local minimisation routine that can find a local minimiser $L(G)$, starting from any initial value $G$.

```
Step_1 Compute $L(G_{init})$. 
$F(G_{prev}) := f(L(G_{init})), F(G_{opt}) := f(L(G_{init}))$;
Step_2 do
  $G_{cur} := \text{perturbation}(G_{prev}), F(G_{cur}) := f(L(G_{cur}))$;
  Generate a random number $r \in (0,1)$;
  if ($F(G_{cur}) \leq F(G_{prev})$) or ($r \leq (\exp(F(G_{prev}) - F(G_{cur}))/T))$ then
    $G_{prev} := G_{cur}, F(G_{prev}) := F(G_{cur})$;
  elseif ($F(G_{cur}) < F(G_{opt})$) then
    $G_{opt} := G_{cur}, F(G_{opt}) := F(G_{cur})$;
  endif
  $T := T \times \alpha$
until (stopping criterion is met)
Step_3 output $L(G_{opt})$ and $F(G_{opt})$ as the best solution and corresponding objective function value.
```

This algorithm has three input parameters. $G_{init}$ is an initial solution. $T$ is the initial temperature. $\alpha$ is a number between 0 and 1.
2.2 Bayesian Networks (BN)

In gene expression profiles, there are usually several underlying uncertainties introduced from experimental sources that are the result of differences in sample preparation and other experiment variables such as non-specific cross hybridisation. Thus we treat gene expression measurement as a probabilistic process and treat each gene with expression values as a random variable. By studying the joint distribution over a set of genes, we gain knowledge of the gene interactions that may reflect true functional linkages.

A BN is an acyclic graph $G$ that can handle experimental ‘noise’ in a principled way. Each node represents a variable and each arc represents probabilistic dependencies. It is used to get a joint probability distribution where $X = \{X_1, X_2, \ldots, X_n\}$. The BN combining $G$ and the joint distribution for each variable, given its parents in $G$, can be expressed as $\langle G, \theta \rangle$ where $\theta_{X_i|\text{pa}(X_i)} = P(X_i|\text{pa}(X_i))$, for each value $X_i$ and the set of $P(X_i)$. $\text{Pa}(X_i)$ represents the set of nodes directly pointing to $X_i$ with edges in $G$. Each $X_i$ follows the Markov Independencies such that each variable $X_i$ is conditionally independent of any non-descendant’s node, given the values of its parents. Together, following the chain rule probabilities and conditional independency property, a BN with a specific joint probability distribution over $X$ can be expressed into the following form:

\[
P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \text{Parent}(X_i)).
\]  

(1)

Given a set of gene expression profiles, studying gene connections can be formulated as a problem of learning the BN from the independent samples of the random variables. The exact Bayesian score is determined by the choice of priors for each $G$. Under the mild assumption of priors probabilities, the scoring metric is consistent. As such, the BN that matches the data best will correspond to a better score than other BNs with high probability. The Bayesian Score Matrix (Heckerman et al., 1995) is widely used as the score of the network $G$ that explains the expression data. The score can be calculated as:

\[
S(G : D) = \log P(G | D) = \log P(D | G) + \log P(G) + C.
\]  

(2)

In equation (2), $D$ is the instances of data in the expression profile, $C$ is a constant factor and $P(D|G)$ is the marginal likelihood averaging the probability value over all possible parameter value combinations to $G$.

The Bayesian score is determined by the choice of priors $P(G)$ and $P(\theta | G)$. The prior can be specified using BDe or Bayesian Information Criterion (BIC). The BDe is based on structure equivalence and decomposition rules while BIC is an information-related asymptotic approximation to BDe (Heckerman et al., 1995). Both scoring metrics incorporate a penalty in order to avoid over-fitting of the data. Since BIC over-penalises complexity compared with BDe, which results in more information loss (Heckerman et al., 1995; Yu et al., 2002), we used BDe in our study and the score of $G$ is expressed as the sum of the dependency scores of each variable and its parents in equation (3).

\[
S_{\text{BDe}} = \sum_{i=1}^{n} \text{Score}_{\text{BDe}}(X_i, \text{Pa}(X_i) : D).
\]  

(3)
Given the specified prior and expression profile, inferring the structure of gene relationships is an optimisation problem of searching $G$ for a minimum score. The structure with the optimal score value represents the optimal network of gene connections that we can infer from the data. We note that in many cases, the network structure represents an instance of a set of graphs with the same posterior score under the condition that they have the same underlying connections and $v$ structures (Neapolitan, 2004). We call the optimal network from the Bayesian framework and possessing the minimum score as the optimal BN. The connections in the undirected graph of the optimal BN reflect the functional linkages among genes.

2.3 Two-Level Simulated Annealing (TLSA)

The inference of an optimal BN, given data and priors, is proven to be NP-complete (Friedman et al., 2000). The number of network structures is super exponential in the number of random variables (Friedman and Koller, 2003). The closed form solutions can be obtained only over domains of small size or with additional ‘prior knowledge’, such as ordering information (Ott et al., 2004). Normally, for the applications with a large number of random variables, such as microarray expression data sets, it is difficult to find the optimal solution for all genes, given the very limited number of samples. A sampling method could be used on large numbers of random variables to extract significant sub-networks over networks with very similar high scores (Friedman et al., 2000). Alternatively, local heuristic search algorithms such as hill-climbing (Rich and Knight, 1991), greedy random walk (Barber and Ninham, 1970) and Simulated Annealing (SA) (Kirkpatrick et al., 1983) are widely used, but on relatively small sets of random variables to produce the optimal solution. The underlying assumption in this case is that the topology of networks with the optimal score can be summarised by the obtained network. Among the heuristic searching algorithms, the SA algorithm has been applied in many applications in the biological domain, such as learning genetic regulatory networks from expression profiles (Hartemink et al., 2002), predicting protein functions from interaction data (Vazquez et al., 2003), RNA secondary structure alignment (Kim et al., 1996) and structure learning from crystal chemical rules (Pannetier et al., 1990). It is an extension of a Monte Carlo method to determine the equilibrium states of a collection of atoms at any given temperature $T$ (Kirkpatrick et al., 1983).

It accepts not only a solution with a better objective function value, but also a solution with a worse objective function value conditionally, with the hope that accepting such a solution can help get out of a local basin. It has been shown to be very successful in solving combinatorial optimisation problems, including Bayesian structure inference (Hartemink et al., 2001, 2002; Ideker et al., 2000; Yu et al., 2002). However, SA suffers from certain drawbacks in the following scenario: after a large number of iterations (the temperature drops to a low degree), the algorithm arrives at a strictly local optimiser that is not a global optimiser. Even after applying a perturbation on the local minimiser (Mokhtar et al., 1993), the new solution is very likely to fall into the same catchment. Since the local minimiser is a strict one and $T$ is low, this move will be rejected. In such a case, the algorithm tries many times to move in vain and stops at a local optimiser.

Here, we apply an improvement over the traditional SA, called TLSA as presented in Algorithm 1. TLSA was able to find an optimal solution in simulations in our previous work (Wang et al., 2004). Now we apply TLSA to infer the functional linkages between gene pairs in the optimal network structure. It differs from the traditional SA in that the
perturbation is performed at one level (the higher level) while the decision is made at another level (the lower level). To use TLSA, we assume that there is a local minimisation routine that can be used to find a local minimiser $L(x)$ from any point $x$. Suppose we make a perturbation $x'$ from the current solution $x$. The traditional SA would compare the value $f(x')$ with $f(x)$ to decide whether to accept or reject this move. In TLSA, however, we compare $f(L(x'))$ with $f(L(x))$ to decide whether to accept or reject this move. This is best illustrated in Figure 1.

**Figure 1** Searching minimiser with TLSA. With the function $f(x)$, for any $x$ which lies between $z_0$ and $z_1$, the local minimisation routine will find the local minimiser $x_1$. For any $x$ which lies between $z_1$ and $z_2$, the local minimiser will be $x_2$. For any $x$ which lies between $z_2$ and $z_3$, the local minimiser will be $x_3$. Suppose that the current solution is near $x_2$ and the temperature is very low. Then the traditional simulated annealing will be trapped at $x_2$. With TLSA, however, it will be very easy to move out of this catchment. This is so because for any small perturbation from $x_2$, the corresponding local minimiser will be the same as the local minimiser from $x_2$. Therefore any such perturbation will be accepted. This will increase the chances of moving to the baseline near $x_3$, which is the global minimiser.

For the function $f(x)$ illustrated in Figure 1, for any $x$ that lies between $z_0$ and $z_1$, the local minimisation routine will find the local minimiser $x_1$. For any $x$ that lies between $z_1$ and $z_2$, the local minimiser will be $x_2$. For any $x$ that lies between $z_2$ and $z_3$, the local minimiser will be $x_3$. Suppose that the current solution is near $x_2$ and the temperature is very low. Then the traditional SA will be trapped at $x_2$, because any small perturbation from $x_2$ will have a function value strictly bigger than $f(x_2)$. With TLSA, however, it will be very easy to move out of this catchment. This is so because for any small perturbation from $x_2$, the corresponding local minimiser will be the same as the local minimiser from $x_2$. Therefore any such perturbation will be accepted. This will increase the chances of moving to the baseline near $x_3$, which is the global minimiser. However, TLSA also has its limitations. For example, if we are near $x_1$ and the temperature is very low, then it will be difficult to move from $x_1$ to $x_2$, as the local minimiser for points in $(z_1, z_2)$ has a larger value than that for points in $(z_0, z_1)$. As a result, it will be difficult to move from $x_1$ to the global minimiser $x_3$, when the temperature is very low. Yet TLSA clearly has advantages over the traditional SA.

In order to apply TLSA, we need to have an efficient local minimisation routine. Here we use the following local minimisation routine. We start with an empty network that has no edge connecting nodes. We search the local minimiser using the following operations: adding an edge, deleting an edge and reversing an edge. Given certain $x_{\text{cur}}$, which is a network structure, there are $n(n - 1)$ network structures in the neighbourhood of $x_{\text{cur}}$ where $n$ is the number of nodes in the network. Therefore, we can apply $O(n^2)$ single step operations to find $x_{\text{opt}}$ in the neighbourhood of $x_{\text{cur}}$. 
3 Results

3.1 TLSA applied on synthetic data

NetSim, a simulation software package written in Java, was used to generate ‘benchmark networks’ and synthetic data sets corresponding to those networks. A benchmark network is a simulated network that is generated from a probability dependency that specifies dependency among variables randomly. If a node A is conditionally dependent on another node B, it is very likely to have a directed edge from A to B. Given a benchmark network, the synthetic data set is generated from the network structure that produces the joint distribution. A network inference algorithm was applied to this data and the accuracy of the TLSA algorithm was evaluated by comparing the recovered networks from TLSA to the original networks. The learning and evaluating process consists of four steps. First, probability dependencies on ten genes are randomly generated. The joint distribution probability values are also generated according to the dependency relationships. We then build a benchmark network based on these gene-dependent relationships. Second, we use the Monte Carlo method in the program Tetrad3 (Glymour, 2001) to generate a 10 × 300 synthetic binary dataset of gene expression profiles. In each dataset we have 10 genes, with 300 experiments associated with each gene. Third, we apply TLSA to search for the optimal network with an optimal score on this synthetic data. The network that corresponds to the optimal score value is the best network that we could learn from the data using TLSA. Fourth, each network that we learn from the TLSA search is compared to its corresponding benchmark network and tested in terms of precision and recall (given Truth Positive value (TP), False Positive value (FP) and False Negative value (FN), recall = TP/(FN + TP) and precision = TP/(FP + TP)). A FP is an edge that is derived from the learning method but does not exist in the real network. A FN is an edge that exists in the real network but is not found by the learning method. A TP means the real edge is found using the learning method. The process on one network is displayed in Figure 2. Dependency and probability distributions for each gene were generated randomly (Table 1). The random gene set size was ten. Figure 2(a) shows the expected GN that is constructed from the values in Table 1. Figure 2(b) shows the network that is inferred using TLSA. In this experiment, TLSA predicted six true edges, three FP edges, and no FN edges. Thus, the recall value for TLSA was 100%, and the precision was 66%.

The same processes were applied ten times independently, generating benchmark networks and ‘learned’ networks from TLSA searches. We then compared the edges in the benchmark networks and inferred networks to calculate precision and recall values. To test the performance of TLSA vs. alternative approaches to network design, we ran these same experiments and compared them to the SA algorithm and K2 algorithm (Kirkpatrick et al., 1983; Cooper and Herskovits, 1992). The comparison is based on the precision and recall values from the same set of synthetic data. The results show that TLSA recovers network structures with both higher precision and recall values compared to these other commonly used methods for network inference (Figure 3).
Figure 2  The performance of the Two-Level Simulated Annealing (TLSA) method on a synthetic data set: (a) the expected ‘benchmark network’ structure constructed from the synthetic data described in Table 1. The resulting network shows genes with edges to other genes (dark green) and orphans with no linkages (light green). The synthetic binary data set is generated via the Mont Carlo method with size of $10 \times 300$ and (b) the network that is inferred using TLSA on the same synthetic data. Dotted lines represent edges not present in the benchmark network (a). In this experiment, recall = 1.0 and precision = 0.66.

Table 1  Description of the inferred linkages on synthetic data (Figure 2(a)). For a gene set with a size of 10, we randomly generate a directed acyclic graph to represent the relationships among the genes. The discrete Bayesian network is then created by assigning the conditional probability value randomly for each gene in the set. The maximum likelihood estimate of each parameter $P(\text{child and parent})$ is the frequency of the child node and its immediate parent node divided by the frequency of parent in the sample.

<table>
<thead>
<tr>
<th>Child</th>
<th>Parent</th>
<th>$P(\text{child} = 0 \mid \text{parent} = 0)$</th>
<th>$P(\text{child} = 1 \mid \text{parent} = 0)$</th>
<th>$P(\text{child} = 0 \mid \text{parent} = 1)$</th>
<th>$P(\text{child} = 1 \mid \text{parent} = 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene1</td>
<td>None</td>
<td>0.06</td>
<td>0.94</td>
<td>0.42</td>
<td>0.58</td>
</tr>
<tr>
<td>Gene2</td>
<td>Gene1</td>
<td>0.33</td>
<td>0.67</td>
<td>0.37</td>
<td>0.63</td>
</tr>
<tr>
<td>Gene3</td>
<td>None</td>
<td>0.83</td>
<td>0.17</td>
<td>0.69</td>
<td>0.31</td>
</tr>
<tr>
<td>Gene4</td>
<td>Gene5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.37</td>
<td>0.63</td>
</tr>
<tr>
<td>Gene5</td>
<td>Gene6</td>
<td>0.68</td>
<td>0.32</td>
<td>0.11</td>
<td>0.89</td>
</tr>
<tr>
<td>Gene6</td>
<td>Gene1</td>
<td>0.41</td>
<td>0.59</td>
<td>0.24</td>
<td>0.76</td>
</tr>
<tr>
<td>Gene7</td>
<td>None</td>
<td>0.31</td>
<td>0.69</td>
<td>0.04</td>
<td>0.96</td>
</tr>
<tr>
<td>Gene8</td>
<td>Gene6</td>
<td>0.29</td>
<td>0.71</td>
<td>0.63</td>
<td>0.37</td>
</tr>
<tr>
<td>Gene9</td>
<td>None</td>
<td>0.43</td>
<td>0.57</td>
<td>0.28</td>
<td>0.72</td>
</tr>
<tr>
<td>Gene10</td>
<td>Gene1</td>
<td>0.28</td>
<td>0.72</td>
<td>0.97</td>
<td>0.03</td>
</tr>
</tbody>
</table>
Figure 3  Comparison of Two-Level Simulated Annealing (TLSA), simulated annealing, and K2 algorithms in terms of precision and recall. The chart shows precision and recall values on the same set of synthetic data for each method. The white bars represent the precision value distribution and the shaded bars represent the recall value distribution. Percentage values are indicated on the Y-axis. Error bars: standard error of the mean.

Score values can be calculated to infer the best network structure learned from a given dataset, and the network structure with the lowest score value fits the data set best. We compared the score values obtained with TLSA, SA, and K2 from the same simulation dataset. The results show that the score value of optimal networks obtained using TLSA is slightly lower than K2, and significantly lower than SA (Figure 4). The comparisons were made on ten networks with 20 nodes (i.e., genes) and ten networks with a size of 40 nodes. As shown in Figure 4, TLSA performs much better than SA in terms of the optimal score value achieved for the recovered networks. While the score values of TLSA and K2 were similar, K2 requires prior knowledge of network structure while TLSA does not, making TLSA more useful for predicting networks comprised of uncharacterised genes.

Figure 4  Comparison of score values between Two-Level Simulated Annealing (TLSA), simulated annealing, and K2 methods: (a) the optimal scores of TLSA, SA and K2 on 20-node-networks and (b) the optimal scores of TLSA, SA and K2 on 40-node-networks. In each comparison, the same set of simulation data and parameters are used. A lower score value represents a more optimal result. For K2, prior knowledge of node ordering is necessary.
Figure 4 Comparison of score values between Two-Level Simulated Annealing (TLSA), simulated annealing, and K2 methods: (a) the optimal scores of TLSA, SA and K2 on 20-node-networks and (b) the optimal scores of TLSA, SA and K2 on 40-node-networks. In each comparison, the same set of simulation data and parameters are used. A lower score value represents a more optimal result. For K2, prior knowledge of node ordering is necessary (continued)

3.2 Two-Level Simulated Annealing (TLSA) applied on experimental data

We next applied TLSA to find genetic linkages using gene expression data from *Saccharomyces cerevisiae*. The data set was taken from supplementary material in Hughes et al. (2000). The data set is comprised of 300 full genome expression profiles obtained from 276 deletion mutants, 11 tetracyclin regulatable alleles of essential genes, and 13 chemically treated *S. cerevisiae* cultures, each compared to a baseline wild-type or mock-treated culture. Given the normalised expression profile, we further discretised the expression levels to two states using a fixed discretisation procedure, accepting data that represents at least a 2-fold change in expression value. After this pre-processing, we arrived at a subset comprised of 553 genes.

Many reports of gene network modelling use as a data set comprised of a small group of genes that are known to be functional linked in the same pathway (Hartemink et al., 2001, 2002; Ideker et al., 2000). Thus, gene selection is based on prior biological knowledge before data preprocessing is applied on relevant microarray data. This of course leads to significant bias because while some genes that have a small expression ratio are usually removed from the data set prior to modelling (e.g., they exhibit < 2-fold change in expression), they may still be artificially included for analysis because they belong to a certain known pathway. The data associated with these genes will not give a strong signal for inferring gene functional linkages and might decrease the precision. In our work, we use a uniform method to select genes without prior knowledge. The K-means clustering method was applied to generate clusters with correlated expression patterns using $k = 10$. TLSA is then applied on each cluster to learn the network structure.

Since only very limited pathway information is known, we sought to use GO for validation of the inferred networks. GO provides gene annotation of cellular components, molecular function and biological processes and is a very powerful tool for finding gene functional links through a common vocabulary. We ranked the resulting clusters according to the number of genes possessing similar functions. Three clusters that had a
majority of genes with known function (i.e., < 10 genes of unknown function) are shown in Figures 5–7. We performed function enrichment analysis based on their hypergeometric distributions. In cluster 1, most of the genes have a biological function in transport (GO: 0006810), establishment of localisation (GO: 0051234) and localisation (GO: 0051179), with a $P < 10^{-5}$. The enriched functions in cluster 2 include purine ribonucleotide biosynthesis (GO: 0009152) and cell wall organisation and biogenesis (GO: 0044264) with $P < 10^{-2}$ and the enriched functions in cluster three include vitamin biosynthesis (GO: 0009110) and water soluble vitamin metabolism (GO: 0006767) with a $P$ value of less than $10^{-6}$. Considering that more than 50% of the genes in clusters 2 and 3 have not been annotated with any biological process, the real statistic significance in terms of $P$ value could actually be much lower. Those genes with no ontological annotation cannot provide information for evaluation, but the linkages with annotated genes might provide putative functional roles for these genes.

We further evaluated these networks using GO information describing the putative functions of these genes within the network. A GO online application, FatiGO (Al-Shahrour et al., 2004), which is able to test the significance of gene linked in a network statistically, was used to group genes that are closely related to the same GO category. In FatiGO, there are five categorisation levels ranging from two to six; the higher the level, the more confidence one has that a pair of genes are annotated in the same biological process. We did not include genes that have an unknown GO category when we calculated the score value of the network. The relationships involving these genes can give us putative relationships, but cannot help us with the evaluation of the network. The total score of the network $N$ using GO information is calculated using the following score function,

$$
\text{Score} = \max_{i,j \in N} L_{i,j} \times P_{i,j},
$$

Figure 5 A yeast genetic network inferred with TLSA visualising the relationships among 19 genes. Nodes represent yeast genes (gene name indicated within each circle). An edge directed from one node to another indicates these two genes are correlated with certain dependency. The network structure is the optimal Bayesian network that is inferred using TLSA.
Figure 6  A yeast genetic network inferred with TLSA visualising the relationships among 38 genes. Nodes represent yeast genes (gene name indicated within each circle). An edge directed from one node to another indicates these two genes are correlated with certain dependency. The network structure is the optimal Bayesian network that is inferred using TLSA.

Figure 7  A yeast genetic network inferred with TLSA visualising the relationships among 43 genes. Nodes represent yeast genes (gene name indicated within each circle). An edge directed from one node to another indicates these two genes are correlated with certain dependency. The network structure is the optimal Bayesian network that is inferred using TLSA.
This function gives the sum of score values for all the genes in the network. \( P_{(i,j)} \) is an integer value and \( P_{(i,j)} \in [2, 6] \). \( i \) and \( j \) are nodes in the network \( N \) and \( L_{(i,j)} \) is defined as follows:

\[
L_{(i,j)} = \begin{cases} 
1, & \text{if } (i, j) \in N \\
0, & \text{otherwise.}
\end{cases}
\]  

(5)

For any edge \((i, j)\) in the network \( N \), there may be more than one \( P_{(i,j)} \) values because category terms are mapped in an hierarchical manner. If a pair of genes is found in the upper level of the hierarchical structure, it is likely that they also appear in the lower level. In that case, \( P_{(i,j)} \) with the highest level value will be chosen in order to maximise the score. The score function in equation (1) gives an analytic measure on the confidence of the function linkages which are inferred from the network. Intuitively, the number of proteins that are associated with a higher level GO term is smaller than that of in a lower level GO term considering the hierarchical structure of GO. In other words, a protein pair, which shares a GO term in a higher level, has higher statistical confidence, in terms of functional linkage. The score for the network is calculated by counting the score for each link in the inferred network. The evaluation for the statistical significance based on equation (1) is described later in the paper. An optimal network derived using TLSA on a cluster with 26 genes is shown in Figure 5. Seven out of 26 genes separated without any functional links to other genes in the cluster, yielding a structure with 19 genes. The results, summarised in Table 2, show that every linkage is supported by GO information; all of these genes are associated with biomolecular metabolism or transport in yeast.

<table>
<thead>
<tr>
<th>Gene A</th>
<th>Gene B</th>
<th>Biological process</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDH1</td>
<td>IDH2</td>
<td>Amino acid metabolism (GO:0006520)</td>
<td>6</td>
</tr>
<tr>
<td>IDH2</td>
<td>CIT2</td>
<td>Glutamate biosynthesis (GO:0006537)</td>
<td>6</td>
</tr>
<tr>
<td>IDH2</td>
<td>PDH1</td>
<td>Organic acid metabolism (GO:0006082)</td>
<td>4</td>
</tr>
<tr>
<td>IDH2</td>
<td>TAF1</td>
<td>Macromolecule metabolism (GO:0043170)</td>
<td>4</td>
</tr>
<tr>
<td>IDH2</td>
<td>ACO1</td>
<td>Glutamate biosynthesis (GO:0006537)</td>
<td>6</td>
</tr>
<tr>
<td>FRE2</td>
<td>SCW11</td>
<td>Cell growth and/or maintenance (GO:0008151)</td>
<td>4</td>
</tr>
<tr>
<td>SIT1</td>
<td>ENB1</td>
<td>Siderochrome transport (GO:0015891)</td>
<td>6</td>
</tr>
<tr>
<td>SIT1</td>
<td>CCC2</td>
<td>Siderochrome transport (GO:0015891)</td>
<td>5</td>
</tr>
<tr>
<td>SIT1</td>
<td>FTR1</td>
<td>Transport (GO:0006810)</td>
<td>5</td>
</tr>
<tr>
<td>CIT2</td>
<td>YOR382W</td>
<td>Physiological process (GO:0007582)</td>
<td>2</td>
</tr>
<tr>
<td>TAF1</td>
<td>ENB1</td>
<td>Cell growth and/or maintenance (GO:0008151)</td>
<td>4</td>
</tr>
<tr>
<td>TAF1</td>
<td>ARN1</td>
<td>Cell growth and/or maintenance (GO:0008151)</td>
<td>4</td>
</tr>
<tr>
<td>SIT1</td>
<td>ENB1</td>
<td>Siderochrome transport (GO:0015891)</td>
<td>6</td>
</tr>
<tr>
<td>ARN1</td>
<td>ACO1</td>
<td>Macromolecule metabolism (GO:0043170)</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2: Biological processes and level-values of connected gene pairs, as measured by the Two-Level Simulated Annealing (TLSA) method. Gene A and gene B in the same row of the table indicates they are connected by an edge in the network structure in Figure 6. Genes with no ontological information are omitted. Column 3 describes the biological process that both genes are involved in. In this case, all processes are related to biomolecular metabolism or transport. Level is obtained by the program FatiGO and is based on a search of a Gene Ontology database.
In order to evaluate the statistical significance of the linkages, we determined whether the score value of an inferred network was significantly higher than that of randomly generated networks. We generated 1000 randomised networks with the same size of the actual network using a Monte Carlo approach. For fair comparison, each node in the random network has the same edge connection properties such that the number of edge degrees is maintained. The score for each random network can be calculated using equation (4), which can then be used to derive estimates of the mean $\mu$ and standard derivation $\sigma$. The $P$ value is calculated from the real network score, $\mu$ and $\sigma$.

We evaluated the recovered networks using the $P$ value. For the cluster in Figure 5, our result reveals the underlying regulated gene pairs with very high confidence (Table 3). It is noted that the network also reveals several important pairs of genes that play known roles in the same biological processes. IDH1 and IDH2 are associated with a main pathway of carbohydrate metabolism, amino acid metabolism, amino acid biosynthesis and glutamine family amino acid metabolism. They are also found in the same cellular component, the mitochondrial matrix (GO:0005759). We applied the same method on the other two groups of genes in Figures 5 and 6. The statistical results are listed in Table 3. We observed that TLSA can build networks comprised of genes that share similar biological processes with very high confidence (i.e., with a low $P$ value). In the second cluster, the $P$ value is slightly higher compared with the $P$ values from the other two clusters. We surmise the reason is because 17 out of 38 genes in this cluster fall into the ‘unknown function’ ontological category, which biases the real significance value when considering those genes and connections. Yet for those edges that connect an uncharacterised gene with a gene of known function, the edges may provide putative functional information for these unknown genes.

### Table 3

<table>
<thead>
<tr>
<th>Cluster number</th>
<th>Size</th>
<th>Score</th>
<th>Mean</th>
<th>Std. value</th>
<th>P value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Figure 5)</td>
<td>26</td>
<td>148</td>
<td>136.25</td>
<td>7.89</td>
<td>0.06</td>
</tr>
<tr>
<td>2 (Figure 6)</td>
<td>38</td>
<td>47</td>
<td>41.99</td>
<td>3.92</td>
<td>0.1</td>
</tr>
<tr>
<td>3 (Figure 7)</td>
<td>43</td>
<td>38</td>
<td>29.28</td>
<td>4.68</td>
<td>0.03</td>
</tr>
</tbody>
</table>

4 Discussion

In this paper, the problem of finding functional relationships among genes is formulated as a problem of finding an optimal solution with the best search score value. We present an improved searching algorithm from SA, TLSA, which can search for an optimal value with less likelihood to be trapped into a local basin. We applied the method to analyse both synthetic data and previously described data of *S. cerevisiae* (Hughes et al., 2000). The results of network inference from synthetic data show that our TLSA method can find networks with high precision and recall compared with other methods. We note that the distance between the optimal score value of an inferred network and the score value
of a benchmark network increases as more genes are included in the analysis. The intuitive interpretation is that the searching space is increasing exponentially with the size of the network, which makes finding the global optimal solution more difficult. In the case where many genes are included as the input to the method, more knowledge could be given as prior into the Bayesian framework to make more accurate predictions. One might also average the posterior model distributions to extract a set of linkages from multiple inferred structures (Friedman and Koller, 2003).

Given a certain scoring matrix such as BDe, TLSA can most effectively find the optimal network structure with an optimal score value. We should note, however, that performance of learning methods such as TLSA depend on factors such as the quality of the data, method of discretisation and complexity of underlying biological processes in which the genes under study play a functional role. Some genes known to be important components of many networks, such as STE12 (a promiscuous transcription factor), were not in our gene list after data preprocessing because the signal was weak, thereby obfuscating its dependencies. Capturing the relationships of genes such as this is an important challenge for future studies.

Here, we applied TLSA on real data sets for finding gene functional linkages. In contrast to many other reports, we did not pre-select genes based on prior biological knowledge but on the preprocessed microarray data alone. GO information was used to validate the linkages of genes in the inferred network, and the results show that TLSA is able to successfully find intra-cluster structure among genes in a statistically relevant fashion. This helps us to verify genetic relationships and also to understand the putative function of previously uncharacterised genes.

We note that the method described here can be configured to include other types of data sources, such as protein–protein interaction data, genomic location data, or transcription factor binding site data, into the Bayesian framework to more accurately infer gene function linkages. Given the asymptotically consistent scoring metric for other biological problems, the method can be easily applied to search the optimal solution that is associated with the maximum or minimum score value. The use of prior knowledge or other machine learning algorithms to select the initial set of genes that are functionally related is an area for future research.

Acknowledgements

The authors wish to thank James Lowey and Dr. Edward Suh for assistance with high-performance computing on the ASU/TGen computer cluster.

References


**Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLSA</td>
<td>Two-Level Simulated Annealing</td>
</tr>
<tr>
<td>BSM</td>
<td>Bayesian Score Matrix</td>
</tr>
<tr>
<td>BDe</td>
<td>Bayesian Dirichlet equivalence</td>
</tr>
<tr>
<td>BN</td>
<td>Bayesian Network</td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>GO</td>
<td>Gene Ontology</td>
</tr>
</tbody>
</table>