From genetics to gene networks:

Gene Network Inference via Structural Equation Modeling in Genetical Genomics Experiments

Bing Liu\textsuperscript{1,2,*}, Alberto de la Fuente\textsuperscript{2,3,*} and Ina Hoeschele\textsuperscript{1,2,§}

\textsuperscript{1}Department of Statistics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061
\textsuperscript{2}Virginia Bioinformatics Institute, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0477
\textsuperscript{3}CRS4 Bioinformatica, Parco Scientifico e Tecnologico, POLARIS, Edificio 1, Loc. Piscina Manna, 09010, Pula (CA), Italy

*These authors contributed equally to this work

§Corresponding author

Email addresses:

BL: bliu@vt.edu
ALF: alf@crs4.it
IH: inah@vbi.vt.edu
Abstract

Background

In Genetical Genomics, a segregating population is expression profiled and DNA marker genotyped. An Encompassing Directed Network (EDN) of causal regulatory relationships among genes can be constructed with expression Quantitative Trait Locus (eQTL) mapping and identification of candidate regulator-target pairs. An EDN is likely to contain cycles or feedback loops and needs to be sparsified. In this work, we implement Structural Equation Modeling (SEM) to sparsify the EDN by producing a set of sub-models containing fewer edges and being well supported by the data.

Results

Typically, SEM has been implemented for at most tens of variables. Based on a factorization of the likelihood and a strongly constrained search space, our algorithm can construct networks involving several hundred genes. Parameters are estimated based on the method of maximum likelihood, and structure inference is based on a penalized likelihood ratio and an adaptation of Occam’s Window model selection. The likelihood function is factorized into a product of conditional likelihoods of individual genes (not contained in a cycle) as in acyclic Bayesian Networks, and conditional likelihoods of subsets of genes that constitute cyclic components. The likelihood of a cyclic component is maximized using a genetic algorithm. The SEM algorithm was evaluated using simulated data having known underlying network topologies. For the simulated networks, the SEM analysis had an average detection power of ninety percent and an average false discovery rate of ten percent. The
algorithm was also applied to a sub-network of an EDN obtained from a yeast data set.

Conclusions

In this contribution, we present an initial evaluation of SEM for gene network reconstruction in the context of Genetical Genomics experiments. In contrast with Bayesian networks, the SEM analysis can reconstruct cyclic networks from steady-state data. Our implementation of SEM permits the reconstruction of networks of several hundred genes, and the linear SEM performs well on artificial data generated with random network topologies and nonlinear ordinary differential equations.

Background

System biologists are interested in understanding how DNA, RNA, proteins and metabolites work together as a complex functional network. The gene network is a projection of such a network onto the gene space [1], in the sense that only relationships between genes are modelled, while the physical interactions between them are mediated through other components. While networks including genes, RNA, proteins and metabolites would be more informative, gene networks are system level descriptions of cellular physiology and provide an understanding of the genetic architecture of complex traits and diseases.

Bayesian Networks are currently a popular tool for gene network inference [2-6]. Bayesian networks use partially directed graphical models to represent conditional independence relationships among variables of interest and can describe complex stochastic processes. They are suitable for learning from noisy data (e.g. microarray
Bayesian Networks are Directed Acyclic Graphical (DAG) models, which cannot represent structures with cyclic relationships. However, gene networks reconstructed based on Genetical Genomics (or other perturbation) experiments are expected and have been found to be cyclic. ‘Transcriptional regulatory networks’ (e.g [7]), which contain an edge between a transcription factor and a target gene only if a physical binding of the transcription factor to the promoter region of the target has been experimentally detected, contain only few cycles. Gene networks are phenomenological networks whose edges represent causal influences. These can be physical binding of a transcriptional regulator to the target promoter, or more complicated biochemical mechanisms (involving signal transduction and metabolism), as there is much genetic regulation beyond transcription factors [1]. Recent papers point to the need for methods that can infer cyclic networks, note the limitation of the Bayesian network approach [8], and show better performance of a cyclic method over a Bayesian network algorithm due to the presence of cycles [9].

An alternative approach to the reconstruction of Directed Cyclic Networks (DCGs) is based on the assumption that a cyclic graph represents a dynamic system at equilibrium [10], including a time dimension produces a causal graph without cycles (DAG), which then can be studied using Bayesian Networks, an approach called Dynamic Bayesian Networks [5, 11]. However, this approach requires the collection of time series data, which is difficult to accomplish, as it requires synchronization of cells and close time intervals not allowing for feedback [12]. Samples at wider time intervals represent near steady state data and hence require cyclic network reconstruction.
Xiong et al. [13] were the first to apply Structural Equation Modeling (SEM) to gene network reconstruction using gene expression data. However, their application was limited to gene networks without cyclic relationships by using a recursive SEM, which has an acyclic structure and uncorrelated errors and is equivalent to a Gaussian Bayesian network. These authors reconstructed only small networks with less than 20 genes. Here, we apply SEM in the context of Genetical Genomics experiments. In Genetical Genomics, a segregating population of hundreds of individuals is expression profiled and genotyped. With expression Quantitative Trait Locus (eQTL) mapping and selection of regulator-target pairs, an Encompassing Directed Network (EDN) of causal regulatory relationships among genes (represented by directed edges) can be constructed [14]. The network is called “encompassing” because it contains regulators with both direct and indirect effects on the same targets which are actually only indirect regulations. Therefore, an optimal sparser network (best supported by the data) is a sub-network of the encompassing network [14].

In this study, we present an SEM implementation to search for a set of sparser structures within the EDN that are well supported by the data. The method is evaluated on simulated data with known underlying network structures and on a real yeast data set. Typically, SEM analyses have included at most tens of variables, but our implementation is capable of reconstructing networks of several hundred genes based on a factorization of the likelihood and a strongly constrained network topology search space.
Results

The algorithm was tested on the simulated data and on a sub-network obtained from an EDN generated in [14], using a real data set from a yeast segregating population [15].

Simulated data analysis

Ten data sets of 300 observations each, with different random network topologies were analyzed. These networks had 100 genes, 100 eQTLs, and on average 148 gene → gene and 123 eQTL → gene edges. Their EDN contained on average 360 gene → gene and 301 eQTL → gene edges. On average 42 genes were involved in one to three cyclic components in each data set, with the biggest cyclic component involving on average 37 genes. The algorithm was run on a multi-processor SGI Origin2000 and took between two and eight hours (total time) per data set with an average of four hours. We report the results in terms of false discovery rate (FDR) and detection power. The FDR is expressed as the number of wrongly identified edges divided by the total number of identified edges. Power is defined as the number of edges correctly inferred as a fraction of the total number of edges in the true network. In Table 1, we compared results obtained using Bayesian Information Criterion (BIC) with penalty term ln(N)*df, and BIC(δ) with penalty term d*ln(N)*df. We used the recommended d = 2* logarithm-of-odds (LOD) threshold / log10(N) [16], and a LOD threshold of 3. The results showed that for the simulated data sets, BIC was not sufficiently stringent for the eQTL → gene edges, with an average power of 99% and an average FDR of 22%. For the gene → gene edges, the average FDR was 8%, with average power of 88%. For the eQTL → gene edges, the average FDR with BIC(δ) was 9%, while the average power was 99%. For the gene → gene edges, with BIC(δ)
the average FDR was only 1%, while the power was reduced to on average 78%. Overall, the algorithm performed well, and the results show that the linear SEM seems to be robust under violation of the linearity assumptions.

(Insert Table 1 here)

While the above results were based on retaining a single, final SEM model, for some of the 10 data sets we allowed the topology search algorithm to follow 20 different, random search paths. This was done to determine whether there were different models (topologies) with the same likelihood (equivalent models; see methods section), and to identify multiple models with the same or nearly the same BIC (or BIC(δ)). These additional networks contain important information that would be missed when searching only for a single network, and they reflect the uncertainty about the true network structure after observing the data. About sixteen very similar final models were obtained per data set. Out of an average of 134 detected eQTL → gene edges, the average number of edges different from the best model was 4.4. Out of an average of 153 detected gene → gene edges, the average number of edges different from the best model was 7.9. The average BIC difference to the best model was 26. The average likelihood difference was 12, while the mean likelihood was 26,969. Two models had the exact same likelihood (and hence were equivalent), while having six different eQTL → gene edges and seven different gene → gene edges. Another four sets of two models had likelihood differences smaller than 1, with on average four different eQTL → gene edges and 7.3 different gene → gene edges.

Yeast data analysis

We performed SEM analysis on a sub-network of an EDN obtained from the yeast dataset [14]. To obtain this sub-network, we started out with 168 genes involved in a
cycle and included all genes connected to the cycle genes by up to three edges, and all the eQTLs associated with these genes. The sub-network obtained had 265 genes, 241 QTLs, 832 gene → gene edges, and 640 eQTL → gene edges. After sparsification using our SEM implementation, the resulted network contained 475 gene → gene edges and 468 eQTL → gene edges. The SEM analysis took about 110 hours or 4.5 days (total time) on the multi-processor SGI Origin2000. Figure 1 shows the network topology, with the dotted edges denoting removed edges.

(Insert Figure 1 here)

Table 2 shows the significant biological function groups of the genes in this network. About 41.6% of these genes are involved in catalytic activity, and another 18% are involved in hydrolase activity. All biological functions in Table 2 are significantly enriched in this network.

(Insert Table 2 here)

Discussion

In this contribution, we present an initial evaluation of structural equation modeling for gene network reconstruction in the context of Genetical Genomics experiments. Previous investigations have used Bayesian networks [2-6], but this methodology cannot reconstruct cyclic networks from steady state data and hence is not an optimal method for gene network inference in Genetical Genomics or systems genetics. Because cycles or feedback loops are expected to be common in genetic networks, it
is imperative to investigate alternative methods such as the one we have presented here. Our implementation of SEM permits the reconstruction of networks of several hundred genes, and our ongoing and future research will improve the efficiency of the current implementation.

One possible way to verify the results of our network inference approach would be to check whether the interactions we find also are present in ‘transcriptional regulatory networks’ (e.g. [7]). However, there is (a lot of) genetic regulation beyond transcription factors [1] and therefore such comparison may not be very insightful. For example, a recent study [9] using gene expression data recovered only 10% of the transcription factor (TF) to target relationships known in E coli, while it found about 3 times as many interactions that cannot be simply explained through TF to regulatory motive sequence binding. The yeast sub-network studied in this paper contains cases of genetic regulation that are beyond transcription factors; these are genes coding mostly for metabolic enzymes (table 2) and communicating with each other probably through metabolic changes and metabolic effects on gene expression. Interactions in gene networks thus may correspond to causal effects mediated through signal transduction and metabolism, which are hidden variables when studying gene expression alone. Due to the ‘phenomenological’ nature [1] (rather than ‘mechanistic’, such as physical binding of transcription factors to regulatory sequences) of gene networks it is not trivial to compare our findings to currently existing knowledge.

Maximum Likelihood is the predominant full-information method for parameter inference in SEM. It is therefore natural to perform a model (structure) search based
on an information criterion that is a function of the maximized likelihoods of two competing models. While BIC and BIC(δ) performed satisfactorily in this study, further research into appropriate model selection criteria for large, very sparse networks is required. There is also concern about the validity of BIC for Bayesian network (and hence SEM) inference [17]. We are currently exploring various improvements of and alternatives to our current SEM structure search, including: (i) a modification of the BIC criterion to incorporate structure priors that prefer sparse structures and, for example, structures in which for a trans-regulation the regulator gene $\rightarrow$ target gene edge and the eQTL $\rightarrow$ target gene edge are both present or both absent; (ii) shrinkage methods instead of searching the discrete model space including penalized maximum likelihood and empirical Bayes; (iii) a full Bayesian analysis via Markov chain Monte Carlo algorithms. We are also investigating an approximation to the SEM analysis of cyclic networks by linear regression (LR), which amounts to ignoring the determinants $|I - B|$ in equation (2) and $|I - B_{cc}|$ in equation (5) and substantially simplifies computation. In this case, we can calculate the marginal likelihood of model $k$, $p(Y \mid M_k)$, exactly without resorting to the BIC approximation. The LR approximation can be used in place of the exact analysis, as an excellent starting point for the exact analysis, or as proposal distribution in the full Bayesian analysis.

Our SEM model can be generalized to include certain types of interactions: those between an eQTL and a regulator gene jointly trans-regulating a target gene and epistatic interactions between eQTL found in the eQTL analysis and hence included in the EDN. With this model, we can solve for $y_i$ and assume a normal distribution for the residuals as in equation (2). Furthermore, we have considered networks with only
causal, directed interactions or regulations. However, two genes may be correlated, but there may be no eQTL information available to determine causation. At least in theory such associations or undirected edges can be incorporated via correlations in the residual covariance matrix $E$ in equations (1) and (2). One can then include these off-diagonal elements in $E$ in the EDN and consider them as potentially present in the model search. However, this approach would pose a computational problem, as the presence of off-diagonal elements in $E$ would hinder the factorization of the likelihood.

Our network inference algorithm was implemented in C++, and the programs are available upon request.

**Conclusions**

In this contribution, we present an initial evaluation of structural equation modeling for gene network inference in the context of Genetical Genomics and Systems Genetics experiments. In contrast with Bayesian networks, the SEM approach can reconstruct cyclic networks, with cycles or feedback loops expected to be common in genetic networks. Our implementation of SEM permits the reconstruction of networks of several hundred genes, and the linear SEM performs well on artificial data generated with random network topologies and nonlinear ordinary differential equations. In Systems Genetics of complex diseases, researchers are interested in reconstructing a causal network of relationships among DNA variants, gene expression profiles and phenotypes associated with a disease complex. The method presented in this article should be well suited for this task.
Methods

Encompassing Directed Network

Expression QTL mapping treats gene expression levels as quantitative traits, and identifies genomic regions causally affecting gene expression levels. It identifies a set of eQTL regions and for each eQTL a list of target genes whose expression profiles are affected. Furthermore, using DNA sequence information, genes located in an eQTL region can be identified as candidate regulators of the targets of that eQTL. Using local structural models, regulator-target pairs are identified for all eQTLs, taking into account that an eQTL may affect a target through cis, cistrans or trans regulation [14]. Then, an EDN is constructed by drawing directed edges from the regulator genes and eQTLs to the target genes. We have constructed an EDN using a Genetical Genomics dataset from yeast [14]. Here, we implement Structural Equation Modeling (SEM) to search within the EDN for a subset of sparser network topologies that are best supported by the data.

Structural Equation Modeling

A Structural Equation Model

SEM has been widely used in econometrics, sociology and psychology, usually as a confirmatory procedure instead of an exploratory analysis for causal inference (e.g. [18-20]). Shipley [21] discusses the use of SEM in biology with an emphasis on causal inference. In general, an SEM consists of a structural model describing (causal) relationships among latent variables and a measurement model describing the relationships between the observed measurements and the underlying latent variables. Any SEM can be represented both algebraically through a system of equations and
graphically. A special case is the SEM with observed variables only, where all
variables in the structural model are observed, therefore there is no measurement
model. Our model is a SEM with observed variables, which can be represented as

\[ y_i = By_i + Fx_i + e_i, \quad e_i \sim (\theta, E) \quad i = 1, \ldots, N \]  

(1)

In this model, for sample \( i \) \( (i = 1, \ldots, N) \), \( y_i = (y_{i1}, \ldots, y_{ip})^T \) is the vector of expression
values of all \( (p) \) genes in the network, and \( x_i = (x_{i1}, \ldots, x_{iq})^T \) denotes the vector of marker
or QTL genotype codes. The \( y_i \) and \( x_i \) are deviations from means, \( e_i \) is a vector of error
terms, and \( E \) is its covariance matrix.

Matrix \( B \) contains coefficients for the direct causal effects of the genes on each other.
Matrix \( F \) contains coefficients for the direct causal effects of the eQTLs on the genes.
The structure of matrices \( B \) and \( F \) corresponds to the path diagram or directed graph
representing the structural model, in which vertices or nodes represent genes and
eQTLs, and edges correspond to the non-zero elements in \( B \) and \( F \). Matrices \( B \) and \( F \)
are sparse when the model represents a sparse network. When the elements in \( e_i \) are
uncorrelated and matrix \( B \) can be rearranged as a lower triangular matrix, the model is
recursive, there are no cyclic relationships, and the graph is a Directed Acyclic Graph
(DAG). If the error terms are correlated (\( E \) is non-diagonal), or matrix \( B \) cannot be
rearranged into a triangular matrix (indicating the presence of cycles), the model is
non-recursive. The corresponding graph to a non-triangular matrix \( B \) is a Directed
Cyclic Graph (DCG).

The \( x_i \) may be fixed or random. In Genetical Genomics experiments, the eQTL \( x_i \) are
random because individuals are sampled from a segregating population. However, the
joint likelihood of the \( y_i \) and \( x_i \) can be factored into the conditional likelihood of the \( y_i \)
given the \( x_i \), times the likelihood of the \( x_i \), and the latter does not depend on any of the network parameters in \( B, F \) and \( E \) and can therefore be ignored. Thus, we only need to assume multivariate normality for the residual vectors.

An important issue in non-recursive SEM or DCG is equivalence. Models are equivalent when they cannot be distinguished in terms of overall fit. For DAGs, algorithms for checking the equivalence of two models or for finding the equivalence class of a given model in polynomial time are available [22, 23]. Therefore, model search is performed among equivalence classes rather than among individual DAGs [24]. An equivalence class discovery algorithm for DCGs, which is polynomial time on sparse graphs [25, 26], is available but there is no algorithm for model search among equivalence classes as there is for DAGs. Two DAG models are equivalent if they have the same undirected edges but differ in the direction of some edges (edge reversal) [27]. Two DCG model can be equivalent even if they differ in their undirected edges [25, 26]. In our case, two models cannot be equivalent under edge reversal, because the directions of the edges are determined by the eQTLs. By using an information criterion for model selection with a penalty for the number of parameters, we prefer the sparser model among two equivalent models that differ in the number of edges. Therefore, equivalence is of less concern in our case. Instead of selection among equivalence classes, we use a model search algorithm that selects multiple models (described below).

*Algorithms for likelihood maximization*

A main concern about using SEM for gene network inference is the severe constraint on the network size when using existing SEM software (e.g. LISREL [28]; Mx [29]). Typical applications of SEM include models with at most tens of variables. No
existing software program can analyze models with a size relevant to genomics (hundreds or even thousands of variables). Even the SEM implementation of Xiong et al. [13], which employed a genetic algorithm, was only applied to small networks of under 20 genes. Here, we implement SEM analysis in the context of Genetical Genomics experiments, where the EDN provides a strongly constrained topology search space, allowing us to reconstruct networks of up to several hundred genes.

The most commonly used estimation method for SEM is the Maximum Likelihood (ML) method. Assuming a multivariate normal distribution of the residual vectors, or $e_i \sim N(0, E)$, the logarithm of the conditional likelihood of the $y_i$’s given the $x_i$’s and given a particular structure is:

$$L(y_1, \ldots, y_N | B, F, E, x_1, \ldots, x_N) = \text{constant}$$

$$+ N \ln(|I - B|) + \frac{N}{2} \ln(|E^{-1}|) - \frac{1}{2} \sum_{i=1}^{N} ((I - B)y_i - Fx_i)^{T} E^{-1} (I - B)y_i - Fx_i)$$

This log likelihood is maximized with respect to the parameters in $B, F$ and $E$.

Alternative models or structures (topologies) were compared using information criteria. Information criteria combine the maximized likelihood with a penalty term to adjust for the number of free parameters, and some also adjust for sample size. The information criteria we investigated include the Bayesian Information Criterion (BIC) [30] and a modification BIC($\delta$)[16].

A non-recursive SEM model can be under-identified, while a recursive SEM is always identified. A model is "identified" if all parameters are independent functions of the data covariance matrix. Under regularity assumptions, an under-identified model can be equivalent to an identified model nested within it [31]. Since we prefer the sparser model, our model selection based on an information criterion should arrive
at identified models (a SEM can be checked numerically for under-identification by computing the rank of the information matrix or by repeated model fitting).

The likelihood function is non-linear in the parameters, and therefore an iterative optimization procedure is required for its maximization. A SEM for hundreds of genes has a large number of unknown parameters, rendering likelihood maximization computationally expensive or even infeasible. Fortunately, the likelihood can be factored into a product of local likelihoods which all depend on different sets of parameters, and which are maximized individually in analogy with Bayesian Network analysis. For directed acyclic graphs, the global directed Markov property permits the joint probability distribution of the variables to be factored according to the DAG [27]. Let $V$ be the random variable associated with a particular node (vertex). The factorization can be represented as

$$p(V_1, V_2, \ldots, V_n) = \prod_{j=1}^{n} p(V_j \mid V(\text{parents of } j), \theta_j),$$

where $V(\text{parents of } j)$ is a vector of $V$'s of the parent vertices of vertex $j$, and $\theta_j$ is the parameter vector of the local likelihood $p(V_j \mid \cdot)$. A network with cyclic components (connected cycles, in which any gene can find a path back to itself through any other genes) becomes acyclic when a set of genes pertaining to the same cyclic component is collapsed into a single node, i.e. $V_j$ represents either an individual gene or the set of genes involved in the same cyclic component. Then $p(V_1, V_2, \ldots, V_n)$ can be factored as above, thereby turning the optimization problem from one of thousands of dimensions into many of much smaller dimensions. For genes that are not involved in a cyclic component, the univariate conditional likelihood of a gene is maximized efficiently using linear regression. For the genes involved in a cyclic component, their joint multivariate conditional likelihood is maximized.
For a cyclic component $c$, $p(V_c \mid V$ (parents of $c$), $\theta_c)$ involves the equations for all genes in cyclic component $c$ from (2):

$$y_{ice} = B_c y_{ie} + F_c x_{ie} + e_{ic} ; \quad e_{ic} \sim (0, E_c) \quad i = 1, \ldots, N$$

where $y_{ie}$ is a vector of expression values in sample $i$ for all genes in cyclic component $c$ and their parent genes, which can be partitioned into sub-vectors $y_{icc}$ and $y_{icp}$ pertaining to the genes in cyclic component $c$ and to their parent genes not in cyclic component $c$, respectively; $B_c (F_c)$ is a sub-matrix obtained from the original $B (F)$ matrix by extracting all rows corresponding to the genes in $c$ and all columns pertaining to these genes and their parents; $x_{ic}$ contains the genotype codes of all eQTL parents of genes in $c$; and $e_i$ is the residual vector for all genes in $c$.

Matrix $B_c$ can be further partitioned into $B_{cc}$ and $B_{cp}$, corresponding to columns pertaining to genes in $c$ and parent genes not in $c$, respectively. Then

$$(I - B_{cc}) y_{icc} = B_{cp} y_{icp} + F_c x_{ie} + e_{ic} ; \quad e_{ic} \sim (0, E_c) \quad i = 1, \ldots, N$$

In Equation (4), $y_{icp}$ is a vector of exogenous variables (variables that do not receive any inputs) just like $x_{ic}$. The likelihood function for this model is then

$$L(y_{icc} \mid y_{icp}, B_{cc}, B_{cp}, F_c, E_c, x_{ie}) = \text{constant} + N \ln(|I - B_{cc}|) + \frac{N}{2} \ln(|E_c|^{-1})$$

$$- \frac{1}{2} \sum_{i=1}^{N} ((I - B_{cc}) y_{icc} - B_{cp} y_{icp} - F_c x_{ic}) E_c^{-1} ((I - B_{cc}) y_{icc} - B_{cp} y_{icp} - F_c x_{ic})$$

$$L(y_{icc} \mid y_{icp}, B_{cc}, B_{cp}, F_c, E_c, x_{ie}) = \text{constant} + N \ln(|I - B_{cc}|) + \frac{N}{2} \ln(|E_c|^{-1})$$

$$- \frac{1}{2} \sum_{i=1}^{N} ((I - B_{cc}) y_{icc} - B_{cp} y_{icp} - F_c x_{ic}) E_c^{-1} ((I - B_{cc}) y_{icc} - B_{cp} y_{icp} - F_c x_{ic})$$

$$L(y_{icc} \mid y_{icp}, B_{cc}, B_{cp}, F_c, E_c, x_{ie}) = \text{constant} + N \ln(|I - B_{cc}|) + \frac{N}{2} \ln(|E_c|^{-1})$$

$$- \frac{1}{2} \sum_{i=1}^{N} ((I - B_{cc}) y_{icc} - B_{cp} y_{icp} - F_c x_{ic}) E_c^{-1} ((I - B_{cc}) y_{icc} - B_{cp} y_{icp} - F_c x_{ic})$$

The likelihood function (5) of the genes in a cyclic component is maximized using a Genetic Algorithm (GA) based global optimization procedure. During the model search, the local likelihood of cycle $c$ needs to be re-maximized with respect to $\theta_c$ only if the set of parents of genes involved in the cyclic component has changed.
GA is a stochastic iterative optimization tool [32-34]. We use GA with real number genome, and each parameter is coded as a real number “gene” located on a “chromosome” (a possible solution). New solutions are generated by selection, crossover and mutation. A scoring function is evaluated for each chromosome and used as a selection criterion for inclusion of that chromosome in the next generation’s population. For the termination criterion, we require both a minimum number of generations to be reached, and the fitness score to converge. GA finds a global or near-global optimum for high-dimensional problems. GA can search a very complex parameter space, and jump out of local optima. Though GA is computationally more expensive than the gradient based methods, it has been shown that GA is more successful for problems with very complex parameter spaces [35, 36]. The GA C++ library GAlib (http://lancet.mit.edu/ga/) was used in our implementation.

In our model search algorithm, for re-maximization of the local likelihood of a cyclic component, we use four types of starting values simultaneously in the initial GA population: Random starting points; starting values obtained from Two Stage Least Squares (2SLS) (described below); starting values equal to the current parameter estimates; and starting values from the current parameter values for all genes except 2SLS estimates for the genes directly affected by the deletion or addition of an edge. We use current parameter values as starting values because we search the model space by removing and adding single or few edges at a time, and therefore most parameter estimates do not change or do not change much. However, the parameter values associated with the gene directly affected by the deletion or addition of an edge can
change considerably and we hence initiated them by 2SLS. Using these starting values greatly increased the efficiency of the GA optimization.

GA evaluates the fit of a chromosome using the objective function, which in our case is the log likelihood function for genes in a cyclic component. With diagonal $E$ matrix, the most computationally demanding part for evaluating the objective functions is the computation of the determinants of matrices $(I-B)$. These matrices are sparse, and determinants are calculated using sparse LU decomposition as implemented in the C library UMFPACK, which applies the Unsymmetric MultiFrontal method for sparse LU factorization [37-40]. Since the patterns of the matrices remain the same for a given structure, symbolic factorization is performed only once, and the result is used by all numerical factorizations for objective functions of that structure.

**Starting values from two-stage Least-Squares**

Two Stage Least Squares (2SLS; e.g, \[19, 41\]) is a computationally efficient parameter estimation method for SEM. The 2SLS estimates are computed based on one portion of the model at a time, while ML estimation takes the entire model into account. Therefore, ML is called a "full information" method, while 2SLS is a "partial information" method, and ML estimates are generally better than 2SLS estimates. However, 2SLS is a non-iterative approach and computationally very efficient. We used 2SLS to generate starting values for the GA optimization of the cyclic components.
In 2SLS, the first step is to create predicted values of $y$ using all of the exogenous variables in the system, i.e. solving the reduced form equations:

$$y_i = (I - B)^{-1}(Fx_i + e_i) = D x_i + v_i$$

where for simplicity $x_i$ contains all exogeneous variables and $y_i$ all endogeneous variables. Estimates of $\Pi$ are obtained from this model by Ordinary Least Squares (OLS) and used to obtain predictions of $y_i (\hat{y}_i)$, which are then used in the original model, or

$$y_i = B\hat{y}_i + Fx_i + e_i; \quad i = 1,...,N$$

Estimates of $B$ and $F$ are then obtained by OLS. 2SLS may not work well for some genes with no suitable instrumental variables. An instrumental variable for prediction of an endogenous variable exists only under certain conditions in cyclic networks (e.g. [42]). These conditions are likely not met for all genes in a network. Only if each gene had a cis-linked eQTL, then the conditions would always be met.

**Network topology search**

The EDN contains $2^d$ sub-models, where $d$ is the number of edges. It is impossible to exhaustively search this space even for EDNs of moderate size. Therefore, we adapt a heuristic search strategy based on the principle of Occam’s Window model selection [43] which potentially selects multiple acceptable models. Let $A$ denote a set of acceptable models, $C$ the set of candidate models, and $K$ the set of models with minimum IC (the model selection criterion). The search algorithm includes a Down and an Up component. The Down algorithm consists of the following steps:

0. Initialize $A$ and $K$ as empty sets, and $C$ as a set containing only the EDN.

1. Select a model $M_j$ in $C$ and move it to $A$. Set $IC_{min}=0$. 
2. Select a sub-model \( M_2 \) of \( M_1 \) by removing an edge from \( M_1 \), and compute the model selection criterion for these two models, \( IC_{12} \).

3. 
   a. If \( IC_{12} < T \) (i.e. model \( M_2 \) is strongly better than \( M_1 \)), then remove \( M_1 \) from \( A \) if \( M_1 \in A \), add \( M_2 \) to \( C \) if \( M_2 \notin C \), set \( K \) to the empty set, and set \( IC_{\text{min}} = -\infty \);
   b. else if \( T < IC_{12} < IC_{\text{min}} \) (i.e. \( M_2 \) is the best among all sub-models of \( M_1 \) considered so far), then set \( IC_{\text{min}} = IC_{12} \), replace the model in set \( K \) with \( M_2 \), and remove \( M_1 \) from \( A \) if \( M_1 \in A \);
   c. else if \( IC_{\text{min}} < IC_{12} < 0 \) (i.e. model \( M_2 \) improves \( M_1 \) but is not strongly better and is not the best among all sub-models of \( M_1 \) considered so far), then
      i. with probability \( w \) (e.g. \( w = 0.20 \), \( w = 0.10 \)) this model is chosen as a candidate model by removing \( M_1 \) from \( A \) if \( M_1 \in A \) and adding \( M_2 \) to \( C \) if \( M_2 \notin C \);
      ii. otherwise take no action.
   d. else take no action.

4. 
   a. If there are more sub-models of \( M_1 \), then go to step 2;
   b. else move the model in \( K \) to \( C \) if it is not already in \( C \).

5. If \( C \) is not empty, go to step 1.

Starting from all models accepted in the Down algorithm, the Up algorithm follows the same steps as in the Down algorithm, except every time an edge that was removed from the EDN is added back into the model. Once the Up algorithm is completed, the set \( A \) contains the set of potentially acceptable models.
For large networks with many removable edges, the original Occam’s Window model selection [43] approach may search a very large model space. In the worst case, it is equivalent to an exhaustive search. Therefore, we imposed a threshold $T$ on the IC. Only if the IC of the sub-model strongly improves over the model it is nested in (IC smaller than $T$), then the sub-model is kept as a candidate model. Otherwise, if no sub-model passes $T$ and the minimum IC is less than zero, then the model with minimum IC is kept as a candidate model. The size of the search space depends on the value of $T$. If $T = -\infty$ and probability $w$ is zero, the algorithm is similar to the Greedy Hill search [24, 44]. If $-\infty < O_{t} < 0$, then the algorithm searches a larger network space and possibly accepts multiple models. Because $T$ requires the sub-model to strongly improve over the model it is nested in, it is likely that the search will accept only one final model. Therefore, probability $w$ in step 3.c.i can be set to a positive value to introduce multiple search paths to be followed.

The model or structure search space is constrained to nested models within the EDN, and additionally, certain edges cannot be removed from the EDN, because their removal would contradict the results from the eQTL analysis. If a gene’s expression profile is found to be influenced by an eQTL, then there must remain a direct or indirect path from the eQTL to that target gene in the network. For example, an edge for cis-regulation of a gene by an eQTL cannot be removed unless the eQTL has multiple cis-candidates, in which case one of the cis-edges needs to remain. In our current implementation, we first sparsified the $F$ matrix (eQTL $\rightarrow$ gene), and then the $B$ matrix (gene $\rightarrow$ gene relations). Alternatively, for example, multiple candidate regulators from the same eQTL may be tested first. Then, an eQTL and its candidate
regulator(s) may be updated jointly. In addition, the eQTL analysis can suggest a sequence of edge deletions (e.g. weaker, possibly indirect effects may be tested first).

Data simulation

To evaluate the performance of the linear SEM analysis for gene network inference, we simulated data with non-linear kinetic functions and cyclic network topology in the context of a Genetical Genomics experiments with 300 recombinant inbred lines. We simulated QTL genotypes using the QTLcartographer software [45] and steady-state (equal synthesis and degradation rates and constant gene expression levels in time) gene expression profiles according to the simulated genotypes with the Gepasi software [46-48] using non-linear ordinary differential equations:

\[
\frac{dG_i}{dt} = V_i \cdot \prod_j Z_j \left( \frac{K_{ij}}{I_j + K_{ij}} \right) \times \prod_k \left( \frac{A_k}{A_k + K_{Ak}} \right) - k_i G_i + \xi_i G_i
\]

where \(G_i\) is mRNA concentration of gene \(i\), \(V_i\) is its basal transcription rate, \(K_{ij}\) and \(K_{Ak}\) are inhibition and activation rate constants, respectively, \(I_j\) and \(A_k\) are inhibitor and activator concentrations, respectively (the expression levels of genes in the network affecting the expression of gene \(i\)), and \(k_i\) is a degradation rate constant. Each gene has two genotypes, and the polymorphism is either located in its promoter region affecting its transcription rate (cis-linkage with \(V = 1\) for one genotype and \(V = 0.75\) for the other), or in the coding region of a regulatory gene changing the basal transcription rates of the target genes by multiplying \(V\) by a factor \(Z\) (\(Z = 1\) for one genotype and \(Z = 0.75\) for the other). Each gene has a 50% probability of having a promoter (cis) or coding region (trans) polymorphism. The error parameter \(\theta_i\) represents the "biological" variance and was sampled from a normal distribution with a mean 0 and a standard deviation of 0.1 each time before the calculation of a steady
state. All other parameters were set to 1. Lastly, we also added “experimental noise” to the generated data at 10% proportional to the variance of each gene’s expression values.

The parameters were chosen so that the estimated heritabilities were close to those found in real data. For a simulated data set, we calculated the heritability of an etrait by dividing the steady state variances simulated without biological and technical noise by the variance simulated with biological and technical errors. The simulated etraits had an average heritability of 56% with 60% of the etraits having heritabilities above 57%. The simulated etraits had somewhat lower heritabilities than the actual etraits in the yeast data set where 60% of the genes had estimated heritabilities > 69% [15], which were calculated as (etrait variance in the segregants – pooled etrait variance among parental measurements) / etrait variance in the segregants. The network topologies were generated as described by [48]. For each generated network we created an EDN by adding links from any node $i$ to node $j$, if node $j$ was no more than two edges separated from node $i$ in the true network. The results are reported as FDR and power using BIC [30] and BIC($\delta$) [16] criteria.

**Authors' contributions**

Dr. Bing Liu worked on the maximum likelihood estimation with genetic algorithms and on the network topology search; she carried out the analysis and drafted the manuscript. Dr. Alberto de la Fuente worked on two-stage least squares estimation,
identification of cycle components and path identification, and generated the artificial data used in this study. Dr. Ina Hoeschele conceived and directed the work.

**Acknowledgements**

We thank Rachel Brem and Leonid Kruglyak for sharing the genotype data with us and for providing the raw data of the spotted microarray experiments at the National Center for Biotechnology Information / Gene Expression Omnibus website, [http://www.ncbi.nlm.nih.gov/geo](http://www.ncbi.nlm.nih.gov/geo). We thank Pedro Mendes for useful discussions about GAs. This work was supported by National Science Foundation cooperative agreement DBI-0211863 and the Virginia Bioinformatics Institute.
References


42. Heise DR: **Causal analysis.** New York: John Wiley and Sons; 1975.


Figures

Figure 1 - Network topology of the yeast sub network

Black edges are gene → gene edges in the sparsified network, and blue edges are eQTL → gene edges in the sparsified network. Red dotted edges are removed gene → gene edges, and green dotted edges are removed eQTL → gene edges. Visualized with Cytoscape [49]

Tables

Table 1 - Results of the SEM analysis on the simulated data

<table>
<thead>
<tr>
<th>IC</th>
<th>Edge type</th>
<th>Model Measure</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIC</td>
<td></td>
<td></td>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18.4</td>
<td>24.3</td>
<td>27.4</td>
<td>17.9</td>
<td>19.5</td>
<td>21.6</td>
<td>20.7</td>
<td>19.0</td>
<td>23.9</td>
<td>22.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>99.2</td>
<td>99.2</td>
<td>100.0</td>
<td>97.5</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.6</td>
<td>7.1</td>
<td>7.6</td>
<td>5.7</td>
<td>8.5</td>
<td>3.8</td>
<td>15.3</td>
<td>9.5</td>
<td>11.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>87.6</td>
<td>89.7</td>
<td>89.9</td>
<td>89.3</td>
<td>89.3</td>
<td>88.4</td>
<td>85.9</td>
<td>85.8</td>
<td>88.7</td>
<td>87.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7.5</td>
<td>7.9</td>
<td>7.7</td>
<td>5.1</td>
<td>8.1</td>
<td>7.1</td>
<td>6.3</td>
<td>14.8</td>
<td>11.9</td>
<td>14.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100.0</td>
<td>100.0</td>
<td>99.2</td>
<td>99.2</td>
<td>100.0</td>
<td>96.7</td>
<td>100.0</td>
<td>98.4</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.8</td>
<td>0.0</td>
<td>1.7</td>
<td>0.0</td>
<td>1.6</td>
<td>3.4</td>
<td>0.0</td>
<td>3.4</td>
<td>1.8</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>80.7</td>
<td>82.2</td>
<td>79.9</td>
<td>78.5</td>
<td>81.2</td>
<td>76.2</td>
<td>77.9</td>
<td>77.7</td>
<td>72.7</td>
<td>71.8</td>
</tr>
</tbody>
</table>

FDR and power (in percent) are given for ten artificial data sets using BIC and BIC(δ) criteria, and separately for the eQTL → gene and gene → gene edges.
Table 2 - Significant biological function groups of genes in the yeast sub-network

<table>
<thead>
<tr>
<th>GO_term</th>
<th>Frequency</th>
<th>Genome Frequency</th>
<th>Probability</th>
<th>Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>catalytic activity</td>
<td>41.6%</td>
<td>26.8%</td>
<td>1.5E-07</td>
<td>AAD14 AAD6 ACO1 AKL1 ALD6 AMD2 APN2 ARA1 ARD1 ARP5 AYR1 BDS1 CIT2 COQ5 COX5B DCP2 DIA4 DLD3 DUS3 ECM40 ERF2 EXG1 FET3 FET5 FRE2 GAB1 GCV3 GPA1 GRX5 HIS4 HIS5 HMG1 HMG2 HO HOS4 ICL2 ILV6 KCC4 KTR1 KTR6 LAT1 LEU2 LSC1 LYS2 LYS4 MAP1 MCM6 MET22 MKT1 MSH2 MSK1 MTQ2 MTR3 NFS1 NOP2 NUC1 NUG1 OST2 OST6 PDE1 PDR12 PHO8 PHO85 PLB2 PMA2 POL1 PPZ1 RAD16 RAD52 RAS1 RCK2 RFC4 RFC5 RHO2 RIB3 RPE1 RPM2 RPO1 SAP4 SCO1 SEN1 SHR5 SKM1 PAH1/SMP2 SPO11 SSA4 SUR1 THR4 TIP1 TOP2 TPS1 TRM7 TRP3 TYR1 TYS1 UBP14 UBP16 UGA2 URA3 WRS1 YAL061W RXT2 YEL077C YER138C YER160C YNL045W NMA111 YOL155C YPT53 YPT6</td>
</tr>
<tr>
<td>hydrolase activity</td>
<td>17.8%</td>
<td>10.5%</td>
<td>0.0026</td>
<td>AMD2 APN2 ARP5 BDS1 DCP2 EXG1 GAB1 GPA1 HIS4 HO HOS4 MAP1 MCM6 MET22 MKT1 MSH2 MTR3 NUC1 NUG1 PDE1 PDR12 PHO8 PLB2 PMA2 PPZ1 RAD16 RAS1 RFC4 RFC5 RHO2 RPM2 SAP4 SEN1 PAH1/SMP2 SPO11 SSA4 TIP1 UBP14 UBP16 RXT2 YER138C YER160C YNL045W NMA111 YOL155C YPT53 YPT6</td>
</tr>
<tr>
<td>transporter activity</td>
<td>9.0%</td>
<td>5.6%</td>
<td>0.01485</td>
<td>AAC1 AGP2 ALR1 AQ1 ATO2 ATR1 COX5B CRC1 DIC1 HXT2 ITR1 KAP114 LPE10 MCH4 MRS11 PDR12 PHO91 PMA2 PDR12 PMA2 SAL1 TAT1 UGA4 YFL054C YMC2</td>
</tr>
<tr>
<td>oxidoreductase activity</td>
<td>7.9%</td>
<td>3.5%</td>
<td>0.00666</td>
<td>AAD14 AAD6 ALD6 ARA1 AYR1 COX5B DLD3 FET3 FET5 FRE2 GCV3 GRX5 HIS4 HMG1 HMG2 LEU2 LYS2 SCO1 TYR1 UGA2 YAL061W</td>
</tr>
<tr>
<td>pyrophosphatase activity</td>
<td>6.8%</td>
<td>3.5%</td>
<td>0.00615</td>
<td>ARP5 DCP2 GPA1 HIS4 MCM6 MSH2 NUG1 PDR12 PMA2 RAD16 RAS1 RFC4 RFC5 RHO2 SEN1 SSA4 YPT53 YPT6</td>
</tr>
<tr>
<td>nucleoside-triphosphatase activity</td>
<td>6.0%</td>
<td>3.2%</td>
<td>0.01405</td>
<td>ARP5 GPA1 MCM6 MSH2 NUG1 PDR12 PMA2 RAD16 RAS1 RFC4 RFC5 RHO2 SEN1 SSA4 YPT53 YPT6</td>
</tr>
</tbody>
</table>

Obtained from the Saccharomyces genome database http://www.yeastgenome.org/.

The columns are: significant GO terms; frequency of the terms in genes submitted; frequency of the terms in the whole genome; a score of significance of the terms in the genes submitted; genes involved in the biological process.
Additional files

Additional file 1 – The yeast sub network

The network can be found online in several file formats at http://www.bioinformatica.crs4.org/Members/alf/bmc/.