

A Mixture Model to Detect Edges in Sparse Co-expression Graphs

Haim Bar^{*1} and Seojin Bang^{†2}

¹Department of Statistics, University of Connecticut

²Computational Biology Department, Carnegie Mellon University

April 5, 2018

Abstract

In the early days of microarray data, the medical and statistical communities focused on gene-level data, and particularly on finding differentially expressed genes. This usually involved making a simplifying assumption that genes are independent, which made likelihood derivations feasible and allowed for relatively simple implementations. However, this is not a realistic assumption, and in recent years the scope has expanded, and has come to include pathway and ‘gene set’ analysis in an attempt to understand the relationships between genes. In this paper we develop a method to recover a gene network’s structure from co-expression data, which we measure in terms of normalized Pearson’s correlation coefficients between gene pairs. We treat these co-expression measurements as weights in the complete graph in which nodes correspond to genes. We assume that the network is sparse and that only a small fraction of the putative edges are included (‘non-null’ edges). To decide which edges exist in the gene network, we fit three-component mixture model such that the observed weights of ‘null edges’ follow a normal distribution with mean 0, and the non-null edges follow a mixture of two log-normal distributions, one for positively- and one for negatively-correlated pairs. We show that this so-called L_2N mixture model outperforms other methods in terms of power to detect edges. We also show that using the L_2N model allows for the control of the false discovery rate. Importantly, the method makes no assumptions about the true network structure.

1 Introduction

Broadly speaking, statistical analysis of ‘omics’ data consists of studying the relative abundance of biological ‘building blocks’, such as genes, proteins, and metabolites. The goal of many studies involving high-throughput data is to identify differential building blocks - those whose abundance levels vary according to the value of some other factor. These factors include, for example, environmental conditions, disease state, gender, or age. To simplify the discussion, we will henceforth use genomics terminology, where the building blocks are genes and the abundance is their expression level. Many biological studies use statistical methods which focus on individual genes and rely on the unrealistic, but mathematically convenient assumption that the expression levels are independent across genes. However, other methods drop this assumption and acknowledge that multiple genes are likely to work as a group associated with the same biological process, thus providing

^{*}haim.bar@uconn.edu

[†]seojinb@cs.cmu.edu

not only more complex functionality, but also robustness to detrimental mutations. A common assumption with this approach is that related genes share the same regulatory process, and therefore, their expression levels are expected to be highly correlated. Thus, the relevant measurements in this context are *co-expression* levels of pairs of genes, which can be measured in terms of Pearson’s correlation coefficient, Mutual Information, Spearman’s rank correlation coefficient, or Euclidean distance. Gene co-expression data can be seen as a network; an undirected complete graph in which nodes represent genes and weights are assigned to edges according to the strength of the association between each pair’s expression levels, across multiple samples.

In order to analyze gene networks, a number of authors define ‘modules’ as sets of genes that have similar expression patterns; they can then focus on a small number of intramodular ‘eigengenes’ or ‘hub genes’ instead of on thousands of genes [Stuart et al., 2003, Zhang and Horvath, 2005, Eisen et al., 1998, Taylor et al., 2009, Barabási et al., 2011]. The weighted gene co-expression network analysis (WGCNA, Zhang and Horvath 2005) is a widely used package that implements this approach. Since genes belonging to different modules are expected to be much less correlated than genes within the same module, it is argued that in statistical analysis which is based on hub genes the aforementioned independence assumption is more reasonable. Thus, one can try to find differentially expressed hub genes with respect to some trait or treatment. Another approach is two-sample test which directly compares covariance matrices between two populations. In particular, two-sample test for high dimensional covariance matrix has been studied [Li et al., 2012, Cai et al., 2013, Schott, 2007]. Recently, Zhu et al. [2017] suggested a sparse leading eigenvalue driven test to compare two high-dimensional covariance matrices obtained from schizophrenia and normal groups and identified novel schizophrenia risk genes. Genes that are found to be highly associated with a trait or a treatment can be further analyzed by using knowledge-based pathway analysis tools such as Gene Set Enrichment Analysis (GSEA, Subramanian et al. 2005, Mootha et al. 2003). Pathway analysis differs from co-expression analysis in that it uses *pre-defined* gene sets in public databases such as the Gene Ontology (GO, Consortium et al. 2004) or the Kyoto Encyclopedia of Genes and Genomes (KEGG, Kanehisa and Goto 2000). One can then test which pathways are over- or under-represented in the identified modules [Khatri et al., 2012].

Such analyses of gene networks require accurate estimates of adjacency or covariance/precision matrices. However, when the dimension of genes is larger than the sample size, it becomes challenging. The problem of estimating large, sparse adjacency or covariance/precision matrices have been thoroughly studied in modern multivariate analysis. Researchers have suggested various approaches using regularization techniques designed for estimating large, sparse matrices. One of most commonly used approaches is to use penalized maximum likelihood. For example, Meinshausen and Bühlmann [2006] proposed to estimate a precision matrix by imposing an l_1 penalty on a Gaussian log-likelihood to increase its sparsity. It uses a simple algorithm to estimate a sparse precision matrix, by fitting a regression model to each variable with all other variables as predictors, and they use the lasso to obtain sparsity. Friedman et al. [2008] suggested a simple but faster algorithm called graphical lasso, which also estimates a sparse precision matrix using coordinate descent procedure for lasso. Yuan and Lin [2007], Banerjee et al. [2008], Rothman et al. [2008], Levina et al. [2008] also proposed algorithms to solve the l_1 penalized gaussian log-likelihood to estimate a large, sparse precision matrix. These methods have been extensively used to estimate sparse gene network. However, the maximum likelihood methods assume the data is independently and identically generated from Gaussian distribution. This may not be a problem for analyzing gene data obtained from (relatively) large population using retrospective study such as the TCGA data [NCI and NHGRI]. However, in contrast, the gene expression data obtained from prospective studies are usually small. For example, an experiment comparing a gene knockout mouse cohort with wild-type cohort usually collects about 10-20 samples for each cohort.

The main goal in this paper is to introduce a new method to uncover the structure of sparse gene networks from such data. That is, we want to detect which pairs of genes are co-expressed (measured in this paper in terms of Pearson’s correlation coefficient.) Using graph-theory terminology, genes are represented by nodes in a graph, and pairs of co-expressed (correlated) genes are represented by edges between the corresponding nodes, with weights proportional to the strength of the correlation between each pair. Typically, gene expression datasets consist of thousands of genes. Its complete graph therefore contains millions of edges, while the true graph, which is very sparse, only contains a small fraction of all possible edges with a vast majority of pairs that are not connected. (or, equivalently, the weight of most edges is zero.) With a finite sample, however, the observed correlations are not zero, even for uncorrelated pairs, so the first problem is to define an accurate decision rule which will be used to differentiate between spurious correlations and true edges. A second, related problem, is how to perform the computation needed to establish such a rule in practice, since the estimated adjacency matrix is large and non-sparse.

To address these challenges, we first obtain weights, w_{ij} , by applying Fisher’s Z-transformation to the (sample) correlation coefficients, r_{ij} . For uncorrelated pairs, the asymptotic distribution of w_{ij} is normal, with mean zero and variance $N - 3$, where N is the sample size. This motivates fitting a mixture model to $\{w_{ij}\}$ in which the majority of pairs belong to a normally distributed ‘null component’, and a small percentage of the weights belong to one of two ‘non-null components’, which follow log-normal distributions (one for positive and one for negative correlations). This so-called L_2N model was first presented in Bar and Schifano [2018a], in the context of identifying differentially expressed (or dispersed) genes. This approach has four advantages. First, the L_2N mixture model leads to shrinkage estimation and to borrowing strength across all pairs, which increases the power to detect co-expressed pairs. Second, the specific form of the mixture model allows us to establish a decision rule which controls the error rate. Third, the mixture model lends itself to a computationally-efficient estimation of the parameters via the EM algorithm [Dempster et al., 1977]. Finally, our method makes no assumptions regarding the underlying structure of the true network. In contrast, methods assuming a specific structure and focusing on modules and eigengenes or hub genes, are only suitable for modular or hierarchical networks where nodes within a module are highly connected but connections across modules are relatively rare. However, biological networks such as protein-protein interaction and gene interaction networks may have other network features where modules cannot be clearly partitioned. Such is the case, for example, with scale-free networks [Ravasz et al., 2002, Wuchty et al., 2003, Tong et al., 2004], a scale-free regime followed by a sharp cutoff [Newman, 2001, Amaral et al., 2000, Jeong et al., 2001], and other networks with curved degree distributions [Zhang et al., 2011, Chu et al., 2012, Smith, 2006, Radrich et al., 2010].

To demonstrate our method, we use the ‘CNV 16p11.2’ gene expression dataset from Horev et al. [2011]. In their paper, Horev et al. [2011] look at the consequences of copy number variations in the p11.2 location of chromosome 16 and show dosage-dependent changes in gene expression, as well as in behavior and brain anatomy. The effects of deletion in 16p11.2 (one copy) are severe, and include increased infant mortality, intellectual disability, and autism spectrum disorders. The effects of duplication of 16p11.2 (three copies) are less severe than deletion. However, duplication has also been associated with autism, schizophrenia, developmental delay, and obesity. The dataset is publicly available through the Gene Expression Omnibus (GEO) database (accession number GSE32012). Our objective is to compare the gene network structure between wild-type mice and mice with either one or three copies of 16p11.2. We focus on the latter since our analysis reveals that there appears to be a substantial and systematic difference between the wild-type and duplication groups in terms of the network structure. We use our method to construct the co-expression gene network in each group and to obtain very sparse graphs. Then, we look for changes in node characteristics between the networks and identify hundreds of genes which form a highly connected

cluster (almost a clique) in the duplication group, but not in the wild-type group. Many of the genes in the highly connected cluster in the duplication group, belong to three sets of functional processes (obtained from a gene ontology (GO) database). One set of genes is involved in sensory perception of smell and in olfactory learning. The second set of genes is involved in processes related to sex determination, single fertilization, and spermatid development and differentiation. The third set of genes is related to pigmentation and to melanin biosynthetic process. We do not observe a similar cluster in the deletion group.

This paper is organized as follows: in Section 2, we present the model to uncover the structure of gene networks and the implementation procedure. We discuss some computational challenges and our proposed solutions. In Section 3, we present results from two simulation studies. In the first simulation study, we evaluate the goodness of fit of our method and its ability to correctly recover the structure. We use four simple network configurations designed to examine performance when we vary the density of clusters and the correlation between a genes in different clusters. In the second simulation study, we compare our method’s ability to correctly identify true edges with that of existing methods. We use five types of network configurations – random, hub, band, scale-free, and overlapped-cluster, and we vary the parameter settings for each configuration. We show that our model successfully eliminates the vast majority of spurious correlations and outperforms other methods in detecting truly connected pairs. In Section 4, we apply our method to the publicly available ‘CNV 16p11.2’ dataset. We find interesting, and we believe novel results, by comparing the gene network structure among differentially expressed genes between wild-type mice and mice with three copies. We conclude with a discussion in Section 5.

2 Statistical Model and Estimation

2.1 A Mixture Model for Edge Indicators in a Gene Network

A gene network can be represented by a weighted, undirected graph in which each node corresponds to a gene and each edge corresponds to a pair of genes which are ‘co-expressed’, meaning that their expression levels are highly correlated. The weights represent the strength of the connection between two genes, namely, their tendency to be co-expressed. Given normalized expression data of G genes, our objective is to discover the network structure – which of the $K = G(G - 1)/2$ pairs are co-expressed. Our general strategy, as described in this section, is to associate with every putative edge in the complete graph with G nodes, a latent indicator variable whose value (0 or 1) is determined by a statistical model.

To start, we define the edge weights in terms of pairwise correlation coefficients. Let \mathbf{x}_g be a vector of normalized expression levels for gene $g \in \{1, \dots, G\}$ obtained from N samples ($N > 3$). Suppose that the true correlation coefficient between genes m and n is ρ_{mn} , and let $r_{mn} = \text{corr}(\mathbf{x}_m, \mathbf{x}_n)$ be the observed correlation coefficient. Using Fisher’s z-transformation, we obtain the estimated weight $w_{mn} = \text{arctanh}(r_{mn})$, which is approximately normally distributed, with mean $\text{arctanh}(\rho_{mn})$ and variance $\frac{1}{N-3}$. Let $E = \{e_{mn}\}$ be the set of true edges in the network. We assume that G is large and that most pairs are not co-expressed, so the network is sparse: $|E| \ll K$. This assumption, along with asymptotic normality of w_{mn} , motivate our model choice. Specifically, we assume that the weights follow the so-called L_2N mixture distribution in Bar and Schifano [2018a]. L_2N is a three-component mixture model in which the ‘null’ component follows a normal distribution with mean 0, representing the majority of pairs that have approximately zero correlation, and the tails (the ‘non-null’ components, for pairs with strong positive/negative

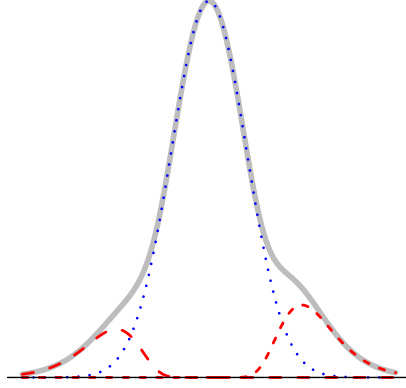


Figure 1: The L_2N mixture model, with probability of the null component (blue dotted curve), $Pr(mn \in C_0) = 0.8$, where mn denotes the edge between nodes m and n in the graph.

correlations) follow log-normal distributions:

$$w_{mn}|e_{mn} \notin E \sim N(0, \sigma^2) \quad (1)$$

$$w_{mn}|[w_{mn} > 0, e_{mn} \in E] \sim \text{LogNormal}(\theta_1, \kappa_1^2), \quad (2)$$

$$-w_{mn}|[w_{mn} < 0, e_{mn} \in E] \sim \text{LogNormal}(\theta_2, \kappa_2^2). \quad (3)$$

Note that σ^2 consists of two variance components, namely, $\sigma^2 = 1/(N-3) + \sigma_0^2$, where $1/(N-3)$ is the variance component due to the asymptotic distribution of Fisher's z-transformed correlation coefficients, whereas σ_0^2 is due to the random effect model, which allows us to account for extra variability among null (uncorrelated) pairs of genes. A graphical representation of the L_2N model is shown in Figure 1.

If we denote the null mixture component in the L_2N model by C_0 , the two non-null components by C_1 and C_2 , the corresponding probability density functions by f_j , and the mixture probabilities by p_j , for $j = 0, 1, 2$, such that $p_0 + p_1 + p_2 = 1$, then we classify a putative edge between nodes m and n in the complete graph into one of the three mixture components based on the posterior probabilities,

$$Pr(e_{mn} \in C_j | w_{mn}) = \frac{p_j f_j(w_{mn})}{p_0 f_0(w_{mn}) + p_1 f_1(w_{mn}) + p_2 f_2(w_{mn})}, \quad j = 0, 1, 2. \quad (4)$$

Let $\mathbf{b}_{mn} = (b_{0mn}, b_{1mn}, b_{2mn})$ be an indicator vector, so that $b_{jmn} = 1$ for the component, j , which has the highest probability, $Pr(e_{mn} \in C_j | w_{mn})$, for the pair mn , and 0 for the other two components. Using this notation, the $K \times K$ matrix $\mathbf{A} = [1 - b_{0mn}]$ denotes the *adjacency matrix* between the G nodes in the graph. Our goal is to obtain an accurate estimate of \mathbf{A} . To do that, we treat the indicators \mathbf{b}_{mn} as missing data, and use the EM algorithm [Dempster et al., 1977] to estimate the parameters of the mixture model. The hierarchical and parsimonious nature of the L_2N model leads to shrinkage estimation and borrowing power across all pairs of genes, as well as to computational efficiency. This is critical, since K is typically very large and can be much larger than the sample size, N . The details regarding the parameter estimation for the L_2N model can be found in Bar and Schifano [2018a].

2.2 Implementation Notes

In the description of the mixture model and the parameter estimation via the EM algorithm, we side-stepped a couple of issues that should be addressed. First, the complete set of pairwise correlation

coefficients, and thus the normalized weights, w_{mn} , cannot be assumed to be mutually independent. Conditionally, they are all asymptotically normally distributed with variance $1/(N - 3)$, but for a fixed m , some w_{mn} will be correlated. Second, obtaining estimates based on all K pairwise correlations is time-consuming and requires storing a very large matrix in the computer's memory, since the values of the indicator variables, \hat{b}_{jmn} , for each pair of genes must be updated in each iteration of the EM algorithm. When the number of genes is large, as is the case in the applications which motivated this paper, we propose taking a random sample of $G' < G$ genes (e.g., $G' = 1000$) and fitting the mixture model to this random subset. Using the smaller subset of genes greatly improves computational efficiency and yields highly accurate estimates. Furthermore, randomly selecting a subset of the genes allows us to assume that the remaining weights used in the (complete data) likelihood function are *approximately* independent. With the estimates obtained from the random subset, it is then possible to compute the posterior probabilities (4) for all K pairs, even on a computer with standard memory capacity.

It was mentioned earlier that the mixture model increases power through shrinkage estimation and is computationally efficient due to its parsimony, but the particular form of the mixture model has an additional benefit: it allows us to estimate how many pairs of genes are correctly (or incorrectly) classified as co-expressed. Specifically, we can decide for each edge, mn , if it is included in the graph based on whether the null posterior probability of w_{mn} is sufficiently small. That is, for a predetermined posterior probability ratio threshold, $T > 1$, we solve

$$\begin{aligned} c_1 &= \arg \min_{w \in (0, \infty)} \frac{\hat{p}_1 \hat{f}_1(w)}{\hat{p}_0 \hat{f}_0(w)} > T \\ c_2 &= \arg \max_{w \in (-\infty, 0)} \frac{\hat{p}_2 \hat{f}_2(w)}{\hat{p}_0 \hat{f}_0(w)} > T \end{aligned}$$

and set $b_{0mn} = 1$ if $w_{mn} \in [c_2, c_1]$, and $b_{0mn} = 0$ otherwise. Alternatively, for some α , we can (numerically) find thresholds $c_1 > 0$ and $c_2 < 0$ such that

$$Pr(\hat{b}_{0mn} \neq 0 \mid b_{0mn} = 0) \approx \hat{p}_0 \int_{-\infty}^{c_2} \hat{f}_0(w) dw + \hat{p}_0 \int_{c_1}^{\infty} \hat{f}_0(w) dw \leq \alpha. \quad (5)$$

That is, we control the estimated probability of a Type I error at a certain level, α . Similarly, we can control the false discovery rate [Benjamini and Hochberg, 1995]. Using the thresholds c_1 and c_2 , we can estimate the probability of a Type II error:

$$Pr(\hat{b}_{0mn} = 0 \mid b_{0mn} \neq 0) \approx \hat{p}_2 \int_{c_2}^0 \hat{f}_2(w) dw + \hat{p}_1 \int_0^{c_1} \hat{f}_1(w) dw. \quad (6)$$

3 Simulation Study

3.1 Data Generated Under the L_2N Model

In the first simulation study, we assess the power and goodness of fit of our model using different configurations, with varying numbers of genes (G), samples (N), degrees of sparsity ($p_1 + p_2$), and graph structures. In this section, data are generated according to the L_2N model from Section 2, and we use different parameters for the log-normal components in the mixture model. We show representative results with $N = 100$ and $G = 500$ (thus, the maximum possible number of edges is 124,750.) Four network configurations are used in this section. They are described in terms of the shape of the $G \times G$ adjacency matrix, A , as follows:

- **complete:** $A = \text{BlockDiag}(J_S - I_S, 0_{G-S})$, where $S = 100$, J is a matrix of 1's, I is an identity matrix, and 0 is a matrix of zeros. This graph contains one clique (complete subgraph) with 100 nodes, and nodes not in the clique are not connected to other nodes. The total number of edges in a graph of this form is 4,950 (about 4% of the possible maximum). $p_1 = 0.0396$, $p_2 = 0$.
- **ar** (autoregressive): A has a Toeplitz structure, with $A_{ij} = 1/(1 + |i - j|)$ if both $i, j \leq S$, where $S = 100$. A graph generated with this AR structure has 4,950 edges. $p_1 = 0.0396$, $p_2 = 0$.
- **two independent blocks:** $A = \text{BlockDiag}(J_S - I_S, J_S - I_S, 0_{G-2S})$, where $S = 50$. A graph generated using this type of an adjacency matrix consists of two distinct cliques, each with 50 genes. The total number of edges is 2,450. $p_1 = 0.0196$, $p_2 = 0$.
- **two negatively correlated blocks:** Similar to the previous configuration, but the two blocks are negatively correlated. The total number of edges is 4,950. $p_1 = 0.0196$, $p_2 = 0.02$.

For pairs i, j such that $A_{ij} = 0$, we generated w_{ij} independently from a standard normal distribution. In the *complete* and *two independent blocks* configurations, for $A_{ij} = 1$ we generated only positively correlated pairs (so $p_2 = 0$), and in the *two negatively correlated blocks* configuration, the pairs were positively correlated within each block, but pairs across the two blocks were generated to be negatively correlated. For the *ar* structure, weights generated from the log-normal distribution appear in the off-diagonals of A in decreasing order. That is, the largest $G - 1$ weights are placed randomly in the secondary diagonal (elements $A_{i,i+1}$), the next $G - 2$ largest weights are placed randomly in the ternary diagonal (elements $A_{i,i+2}$), etc. (Note that the values A_{ij} in this case are only used to indicate that elements along diagonals are equal, but these values are not used to generate the weights.)

All four configurations are sparse, with only 2-4% of the putative edges being present in the graph. The *two negatively correlated blocks* structure has the same sparsity as the *complete* and *ar* graphs, but it has different weights of non-null mixture components. The *ar* graph has a more constrained structure, with weights among pairs of nodes which decay as $|i - j|$ increases, for $i, j \leq N$. Recall, however, that our method does not rely on any assumptions about the structure of the graph. Therefore, it is expected that its performance will only depend on the parameters involved in the mixture model. In the simulations presented here we examined the power of the method to detect edges in the graphs as a function of the location parameters of the log-normal components. We varied θ_1 , such that $\theta_1 \in \{-1.25, -1, -0.75, \dots, 0.75\}$, and set $\kappa_1^2 = 0.25$. In the *two negatively correlated blocks* configuration, we used $\theta_2 = \theta_1$ and $\kappa_2 = \kappa_1$. The weights which were generated according to the L_2N model were transformed into correlation coefficients using the tanh function, and the resulting covariance matrix was used to simulate 500 gene expression values for 100 subjects. We generated 20 different datasets for each configuration, with the results reported in terms of the means of the 20 replicates.

Using the simulated datasets, we applied our method and checked the goodness of fit of the mixture model and the ability to correctly recover the structure of the network, in terms of the number of true- and false-positive edges. In our simulations, we used the approach described in Section 2 to control the false discovery rate at the 0.01 level. To demonstrate how well our algorithm estimates the true mixture model, we plotted for each configuration the histogram of the observed w_{mn} and the fitted mixture distribution and measured the goodness of fit in terms of the root means squared error (rMSE). In all configurations, the root means squared error was very small (≤ 0.01), and the mixture weights (p_0, p_1, p_2) were estimated very accurately and with increasing

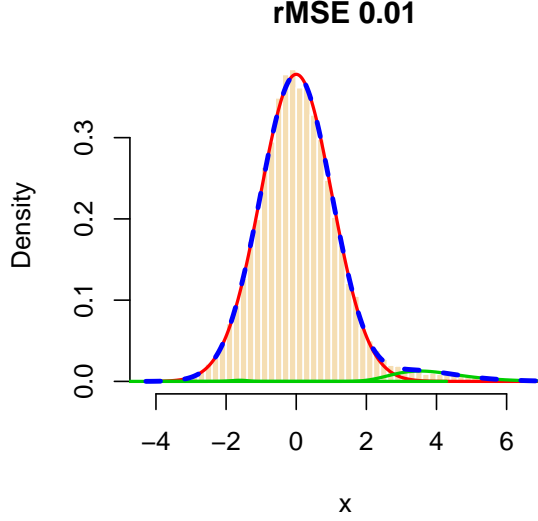


Figure 2: The distribution of $w_{mn} = \text{arctanh}(r_{mn})$ for a simulated data set with 500 genes, of which 100 form a complete subgraph. The total number of edges in the graph is 4,950 (out of a total of 124,750 possible edges.) The red curve represents the null component, the green curves represent the non-null components, and the dashed blue line represents the fitted mixture distribution.

accuracy as θ_1 gets larger. See, for example, Figure S1 in the supplementary material, where the average estimate of p_1 is plotted versus θ using the *two negatively correlated blocks* configuration. A representative goodness of fit plot is shown in Figure 2, for the *complete* configuration, with $\theta_1 = -0.25$. The red curve represents the null component, the green lines represent the non-null components (in this case, C_2 is estimated, correctly, to have a weight which is very close to 0), and the dashed blue line is the mixture distribution.

Arguably, more important than assessing goodness of fit, is determining a method’s ability to recover the true structure of the network correctly, i.e. identify as many existing edges as possible while maintaining a low number of falsely-detected edges. Figure 3 shows the average power of our method (across 20 replications of each configuration) to detect true edges for a range of values for θ_i (with a fixed κ^2) and for different network configurations, where power is the total number of true-positives divided by the total number of edges in the graph. It can be seen that, as the location parameter of the log-normal distribution increases, the power increases and approaches 1. When the data are generated under the L_2N model, this is the expected behavior, since as θ_1 (θ_2) increases, the positive (negative) non-null component, C_1 (C_2), is pushed further to the right (left), making it easier to discriminate between non-null and null components. Note that our method has approximately the same power curve for all four configurations.

The average false discovery rate across all configurations and replications is 0.008. For small values of θ_1 (< -0.75), the average FDR is slightly higher (approximately 0.012) and for $\theta > 0$, the average FDR is less than 0.01. More detailed results regarding the achieved false discovery rate are shown graphically in Figure S2.

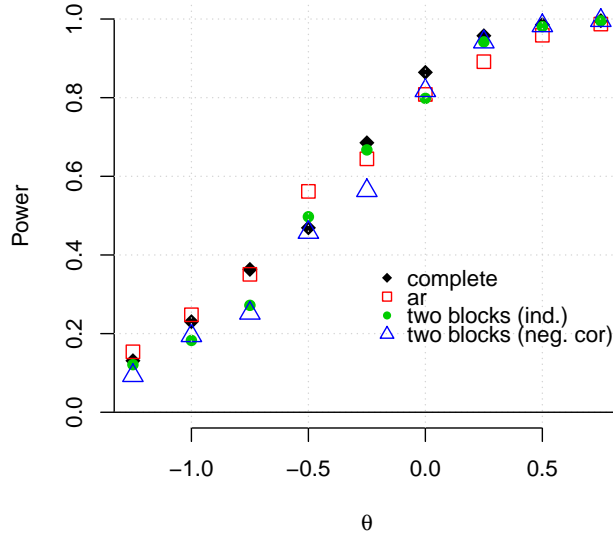


Figure 3: The average power of our method when the data are generated according to the L_2N mixture model, as a function of θ , for a simulated data set with 500 genes and four different forms of adjacency matrices (complete, autoregressive, two independent blocks, and two negatively correlated blocks). In all cases, the FDR was controlled at the 0.01 level.

3.2 Comparison with Existing Methods

In the second simulation study, we compare the ability of our method to recover the true network structure with that of other methods in the literature. For a fair comparison, the data are not generated according to L_2N model. Rather, we use the R-package **huge** [Zhao et al., 2015] where data are generated from a multivariate normal distribution whose covariance matrix Σ is defined as a function of an adjacency matrix A and two additional parameters v and u . We generate five types of network configurations as follows:

- **random**: each edge of a random network is randomly included in the graph using K i.i.d. Bernoulli(p) draws, which results in about $1000 \times (1000 - 1) \times p/2$ edges in the network. In our simulations, we use $p = 0.01, 0.05$, and 0.1 .
- **hub**: a hub network consists of g disjoint groups, and nodes within each group are only connected through a central node in the group, which results in $1000 - g$ edges in the network. We use $g = 25, 50, 100$.
- **band**: an edge, e_{mn} , corresponding nodes $m \neq n$, is set to exist in the graph if $1 \leq |m - n| \leq g$, which results in $(2000 - 1 - g) \times g/2$ edges in the network. We use $g = 5, 25, 50$.
- **scale-free**: scale-free networks are generated using the Barabási-Albert algorithm [Barabási and Albert, 1999], which results in 1,000 edges in the graph.
- **overlapped-cluster**: an overlapped-cluster network consists of g groups, and edges in each group are randomly generated with probability p . We use $p = 0.3$ for $g = 25$ and 50 , and $p = 0.6$ for $g = 100$. The groups are aligned in an adjacency matrix of the overlapped-cluster network; each group shares 20% of the nodes with its left-adjacent group and another 20% with

its right-adjacent group, which results in about $(0.8 \times (g-1) + 1) \times (1000/g) \times (1000/g-1) \times p/2$ edges in the network.

The random, hub, band, and scale-free networks were generated directly by the **huge** package, and we slightly modified a function that generates *cluster* networks from the **huge** package to generate the overlapped-cluster network.

Once each of these five network configurations as well as the appropriate adjacency matrix A are generated, the **huge** package creates the true covariance matrix, Σ . To create Σ , **huge** uses two additional parameters, which allow to control the magnitude of partial correlations: v is a number by which the off-diagonal elements of the precision matrix, Σ^{-1} , are multiplied, and u is a positive number added to the diagonal elements of the precision matrix. We use the default setting: $v = 0.3$ and $u = 0.1$. The gene expression profiles are generated from multivariate normal distributions with mean zero and covariance Σ .

For each network configuration, we generated gene expression profiles for $G = 1,000$ genes, from $N = 70$ samples. We compare our method with three existing methods: Meinshausen-Bühlmann graph estimation (**MB**, Meinshausen and Bühlmann 2006), graphical lasso (**glasso**, Friedman et al. 2008), and **correlation thresholding** graph estimation. Note that MB and glasso identify a group of neighboring nodes for each node in the network. That is, a node m may not be a neighbor of n even though the node n is identified as a neighbor of m . Therefore, we use two different approaches to identify edges based on MB and glasso. First, an edge e_{mn} between node m and n is estimated to exist (i.e. $\hat{A}_{mn} = 1$) if the method chooses the node m as a neighbor of n **and** the node n as a neighbor of m , which we name as MB-AND and glasso-AND respectively. Second, an edge e_{mn} is estimated to exist if the method chooses the node m as a neighbor of n **or** the node n as a neighbor of m , which we name as MB-OR and glasso-OR respectively.

Sparsity levels of the estimated network for MB and glasso are determined by a regularization parameter λ . We use 20 levels for the regularization parameter that decrease on a logarithmic scale from the value, resulting in a network with no edges. The correlation thresholding method is a very simple approach which applies a threshold to the empirical covariance matrix to obtain a sparse estimate of the covariance matrix. We use 30 levels for the correlation threshold, which makes the sparsity level of the estimated network increase evenly from 0 to three times the true sparsity level.

We compare the performance of the methods in terms of the number of true positives (i.e. correctly identified edges) *given the **same** total number of edges identified*. We define the true network in two different ways: (1) based on the adjacency matrix — a pair m, n is connected if and only if $A_{mn} = 1$, and (2) by applying a threshold to the true covariance matrix — for a predetermined t , a pair is connected iff $|\Sigma_{mn}| > t$. The threshold is set to achieve the same sparsity level as that of the adjacency matrix. Since the results from both versions are quite similar, we present the results from the adjacency matrix in this paper and include the results from the true covariance matrix in the supplementary material (Figure S3).

Figure 4 depicts the numbers of true positive edges given the total number of edges identified by each method. Our method (solid red lines) outperforms all the other methods in random, hub, band, and overlapped-cluster networks (see Figure 4A-L). For example, in the hub, $g = 100$ configuration (panel F), the true number of edges is 900 (as indicated by the vertical line), and when our method detects 853 edges, 595 of them are true edges. In contrast, the thresholding method yields approximately 345 true edges out of the total 856 detected; MB-OR and glasso- yield approximately 370–380 true edges out of 900–960 edges detected; and MB-AND yields 312 true edges out of 930 edges detected. Notably, MB and glasso are comparable to, but in some cases worse than the correlation thresholding method in all network configurations.

The black dotted line in the plots represent the expected number of true positive edges when the

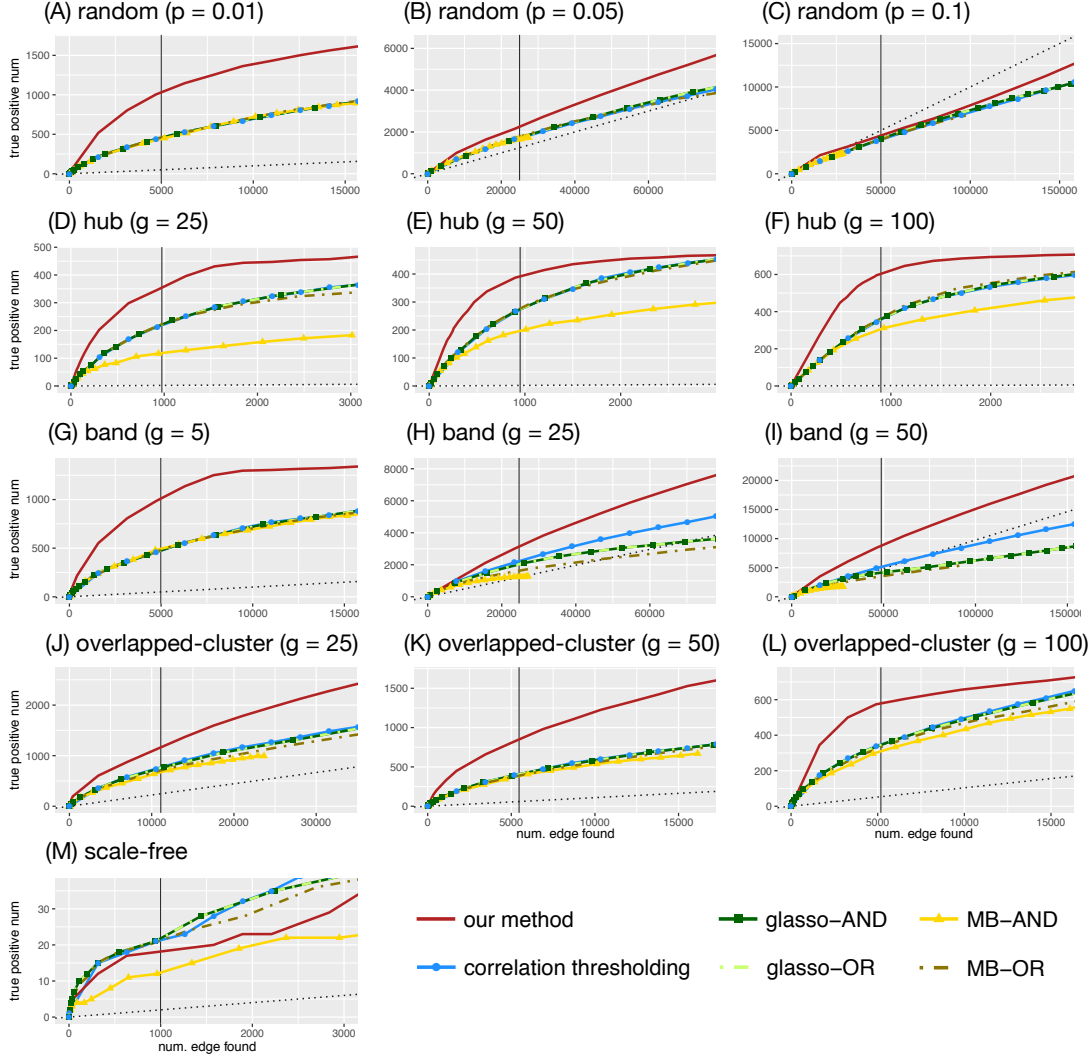


Figure 4: The numbers of true positive edges given the total number of edges identified by each method. The adjacency matrix used by the `huge` package is used to determine the true edges. The y-axis represents the number of true positive edges and the x-axis represents the total number of edges identified. The vertical line represents the number of true edges. The black dotted line is a regression line with 0 intercept and slope equal to the true sparsity, which represents the expected number of true positive edges when the edges are identified in a random manner (uniformly).

edges are identified in a random manner. In the band, $g = 50$ configuration (panel I), the competing methods do as well as or worse than chance, whereas our method gives much better results. In the random, $p = 0.1$ configuration (panel C), the other methods perform worse than choosing edges randomly and our method performs slightly better than chance to a certain point (approximately 30,000 total edges being detected) before deteriorating, but still gives better results than the other methods.

The scale-free configuration appears to be especially challenging for all the methods. They all detect only approximately 10–20 true edges when the total number of edges detected was the 1,000 (Figure 4M). In this case, the correlation thresholding, glasso- and MB-OR perform slightly better than our method.

We further explore scale-free networks with additional network sizes ($D = 200, 500, 2000$). Consider that sparsity levels change as we vary the network sizes: larger D results in a more sparse network. Figure S4 depicts the numbers of true positive edges given the total number of edges for various sizes of scale-free networks. When $D = 2000$, our method is much better than MB-AND (as we have also seen in Figure 4M), and slightly better than the other methods. However, when $D = 200, 500$, our model outperforms the other methods. For example, when $D = 200$, our model identifies around 45 true edges out of 200 edges detected, while glasso-, MB-OR, and the correlation thresholding methods identify less than 30 true edges, and MB-AND identifies around 10 edges.

4 Case Study - copy number variations (CNV) in 16p11.2

Horev et al. [2011] use a mouse model in order to investigate the relationship between copy number variations in the p11.2 location of chromosome 16 and some developmental and neurocognitive disorders in humans. In general, they note that CNVs in 16p11.2 have an effect on brain anatomy and behavior, and that deletion and duplication have opposite effects. They also note that the effects of deletion are more severe than those of duplication and, more specifically, that many of the mice with one copy (16p11.2 deletion) die shortly after birth. In humans, 16p11.2 deletion has been linked to forms of intellectual disability and autism spectrum disorders.

Horev et al. [2011] generated mice with one copy (16p11.2 deletion) and mice with three copies (16p11.2 duplication) and obtained mRNA intensities of 35,557 genes from 37 samples with $n_W = 15$ in the wild-type group (two copies), $n_L = 10$ in the ‘deletion’ group, and $n_P = 12$ in the ‘duplication’ group. One gene has missing data and has been removed from our analysis. The gene expression data for this experiment is available through the National Center for Biotechnology Information (NCBI) website, at <https://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS4430>

Our goal in this section is to demonstrate how to use our model to compare the structure of the gene network (defined by co-expression data) in the wild-type group, with each of the other groups. In the following, we only present results from the comparison between the wild-type and the duplication groups, denoted, respectively, by W and P.

First, we restrict the set of genes to ones which were found to be differentially expressed (DE) between the wild-type group and the duplication group. To detect DE genes, we use the **DVX** method [Bar and Schifano, 2018b] and control the false discovery rate [Benjamini and Hochberg, 1995] at the 0.05 level. The total number of DE genes is 3,454, and the total number of possible edges in the corresponding graphs is 5,963,331.

For each group, we calculate the pairwise correlations for the DE genes and obtain the Fisher’s z-transformed estimates of the weights. To each set of weights, $\{w_{mn}^W\}$ and $\{w_{mn}^P\}$, we fit the L_2N mixture model (Section 2). The mixture distributions fit very well, as measured by the root MSE (for both, $\text{rMSE} \leq 0.01$. See goodness of fit plots Figure S5A and B in the supplementary material).

Table 1: Fitting the L_2N model to the weights in the graphs of the wild-type (W) and duplication (P) groups. \hat{p}_1 and \hat{p}_2 are the estimated probabilities of the positive and negative pairs of genes, respectively. The total number of nodes is 3,454 and the number of possible edges is 5,963,331. Power and FDR are estimated under the L_2N model.

Group	\hat{p}_1	\hat{p}_2	Power	FDR	Edges
Wild-type (W)	0.052	0.009	0.250	0.047	97,556
Duplication (P)	0.064	0.004	0.266	0.049	116,392

Recall that under the L_2N mixture model, it is possible to control the false discovery rate, and using the cut-off points (c_1 and c_2) that were derived in Section 2, we can estimate the achieved power when using c_1 and c_2 . Results from fitting the L_2N model to the two groups are provided in Table 1. Both networks are sparse, with approximately 10^5 edges out of the possible $\approx 6 \times 10^6$. In both networks the number of negatively correlated genes is small, and the majority of the edges correspond to positively correlated pairs. In both cases, the estimated power is similar, but more edges are detected in the P group. The number of edges in common is 37,713. Since we are interested in differences between the two networks, we eliminate nodes which have exactly the same edges in both graphs and end up with 3,005 nodes.

We now consider two network characteristics of nodes, namely the degree, d_m , and the clustering coefficient, γ_m , where m indicates a node m . Let $N_m = \{n \mid A_{mn} = 1\}$ be the set of nodes adjacent to node m , and let $E(N_m)$ be the set of edges between nodes in $N(m)$. Then

$$d_m = |N_m|$$

$$\gamma_m = \frac{|E(N_m)|}{d_m(d_m - 1)/2}$$

If $d_m \leq 1$, the clustering coefficient is defined as $\gamma_m = 0$. By definition, $\gamma_m \in [0, 1]$. The degree of a node is interpreted as the involvement of the node in the network and the clustering coefficient as the connectivity among neighbors of the node. Note that $\gamma_m d_m$ is, by definition, proportional to $|E(N_m)|/(d_m - 1)$, so it is interpreted as (approximately) the average degree among the neighbors of node m . Note also that $\gamma_m d_m$ is bounded by d_m .

Therefore, to visualize and compare the properties of the two networks, we plot $\gamma_m d_m$ versus d_m in Figure 5. The yellow triangles represent the wild-type group, and the blue diamonds represent the duplication group. A couple of things are worth noting about the plot. First, we observe that there is an increasing, linear trend in the wild-type group, which means that as the degree of a node increases, the average degree of each of its neighbors increases. The slope in this case is approximately 0.6, shown as the dashed orange line in Figure 5. This linear relationship suggests that the network does not have a random structure. However, a scale-free structure, which implies that the clustering coefficient does not depend on the degree of a node, is consistent with the plot of the wild-type group. Second, although in general the duplication group has an increasing trend, genes with a large degree (e.g., $d_m > 250$) tend to have a much larger clustering coefficient. This suggests the existence of large, highly-connected cluster(s) of genes. It is also depicted in Figure 6, which shows a bitmap of the data, where a black dot represents an edge between a pair of nodes and white dots represent pairs which are not connected. The genes are ordered according to their structure in the duplication group. The dense block in the bottom-right corner of the duplication bitmap shows a group of more than 350 genes which together form a highly connected cluster. The same group of genes is only sparsely connected in the wild-type graph.

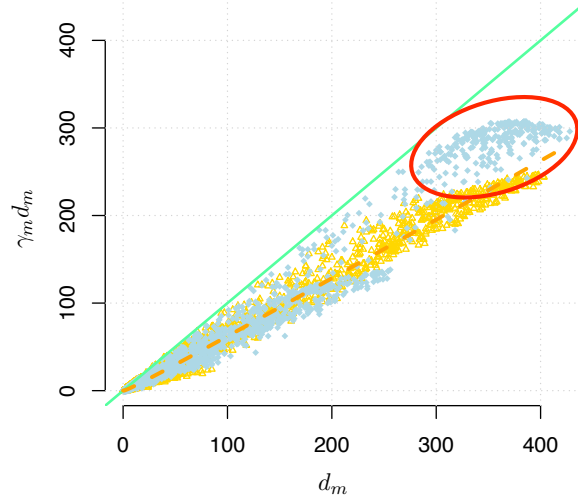


Figure 5: Plot of $\gamma_m d_m$ versus d_m , where γ_m is the clustering coefficient and d_m is the degree of node m . The yellow triangles represent the wild-type group and the blue diamonds represent the duplication group. The orange dashed line is a regression line of the wild-type group with slope = 0.6. The green straight line is the identity line.

Next, for the large cluster of highly connected genes that are depicted as blue diamonds in the red circle of Figure 5, we try to see whether they can be characterized based on their gene ontologies (GO). From the dataset, we extract the ‘GO Process’ for every gene. Note that many genes are not associated with a ‘GO Process’ in this dataset. Other genes may be associated with multiple processes. In the entire dataset (of 3,005 genes), the total number of unique processes is 3,152, while in the subset of highly-connected genes indicated by the red circle in Figure 5 which consists of 334 genes, there are only 55 processes. Among these 55 processes, we find three groups which tend to be connected to a much larger number of genes in the duplication group than in the wild-type. One group consists of sensory perception of smell (see Figure 7, left) and olfactory learning. The second group consists of binding of sperm to zona pellucida, cellular response to testosterone stimulus, sex determination (see Figure 7, middle), single fertilization, spermatid development, and spermatid differentiation. The third group consists of developmental pigmentation, melanin biosynthetic process (see Figure 7, right), melanosome organization, and positive regulation of the melanin biosynthetic process. In Figure 7, the dark-blue diamonds depict the values of $(d_m, \gamma_m d_m)$ for genes in the duplication graph for each process. The red triangles depict the values of $(d_m, \gamma_m d_m)$ for the same genes, in the wild-type graph. The genes from these GO processes tend to belong to large, and highly connected clusters in the duplication group. Whether this is biologically significant remains to be seen in future experiments.

In the comparison between the wild-type and deletion groups (not shown here) we find that both groups have very similar clustering coefficients, and the slope of $\gamma_m d_m$ when plotted vs. d_m is approximately 0.6. However, the deletion group network is more sparse than the wild-type network, and we do not observe a large, highly connected cluster of genes like we did in the duplication group.

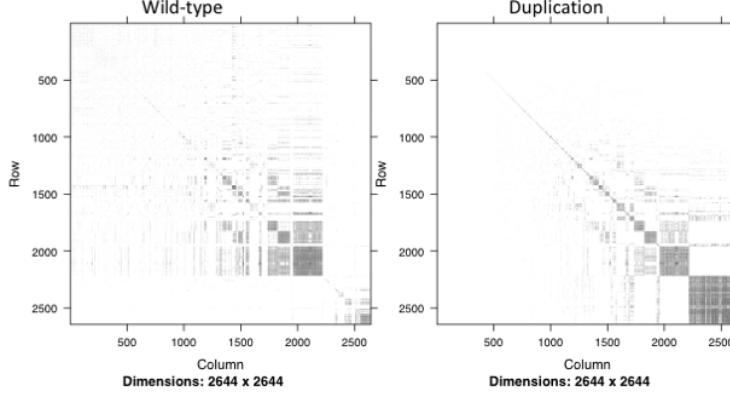


Figure 6: A bitmap of the two networks (Wild-type on the left and duplication on the right.) A black dot represents an edge between a pair of nodes, and white dots represent pairs which are not connected. The genes were ordered according to their structure in the duplication group.

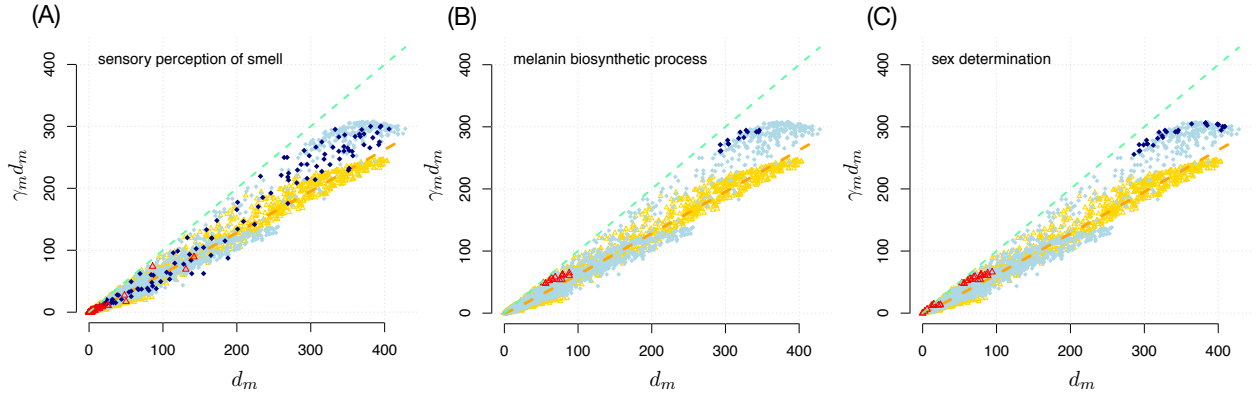


Figure 7: Plots of $\gamma_m d_m$ versus d_m , for two groups of genes, based on the Gene Ontology processes (left - perception of smell, middle - sex determination, right - melanin biosynthetic process). The yellow triangles represent the wild-type group, and the blue diamonds represent the duplication group. The red triangles (wild-type) and the dark-blue diamonds (duplication) represent the genes in the selected gene ontology processes.

5 Discussion

We propose a new approach for detecting edges in gene networks, based on co-expression data. We consider the entire set of genes as a network in which nodes represent genes and weights on edges represent the correlation between expression levels of pairs of genes. We start by modeling the normalized pairwise correlations as a mixture of three components - a normal component with mean 0, representing the majority of pairs which are not co-expressed, and two non-null components, modeled as log-normal distributions, for positively and negatively correlated pairs.

From a theoretical point of view, the so-called L_2N model has the advantage that the overlap between the null component and the non-null components around 0 is negligible. This helps to avoid identifiability problems which are known to affect other mixture models which rely only on normal components, such as the spike-and-slab or a three-way mixture model. Furthermore, the mixture model allows us to accurately estimate the proportion of spurious correlations among all pairs of genes and to derive a cutoff criterion in order to eliminate the vast majority of the edges in

the graph that correspond to uncorrelated genes. We also derived estimators for the probabilities of Type-I and Type-II errors, as well as the false discovery rate associated with the removal of edges from the graph according to the cutoff criterion.

From a practical point of view, this model appears to fit co-expression data extremely well, even when the data are not generated according to the mixture model. Our simulations show that it outperforms other methods in terms of its power to detect edges, and that it maintains a low false discovery rate. Estimation of the model parameters is done very efficiently, using the EM algorithm. In typical gene expression data sets which consist of thousands of genes and millions of edges in the gene network, computational efficiency is critical.

Note that our approach does not require any assumptions about the underlying structure of the network. We only assume that the normalized correlations follow the L_2N model. This is a very modest assumption since the Fisher-transformed correlations are indeed (asymptotically) normally distributed for all the uncorrelated pairs.

Our case study yielded an interesting and, to the best of our knowledge, novel result. It appears that duplication in 16p11.2 is associated with the formation of large, highly connected clusters of genes in very specific gene ontology processes (namely, perception of smell, sex determination, and melanin-related processes.) We hope to establish a collaboration with the group which contributed the original data and investigate this further in future experiments.

We plan to extend this method to handle time varying networks. This will be particularly useful when analyzing gene expression data from repeated measures designs. We also plan to extend the model to applications that involve multiple platforms, such as methylation, proteomics, etc. In principle, with the appropriate normalization technique for each platform, one can simply construct a graph with $G_1 + \dots + G_k$ nodes, where G_i is the number of ‘building blocks’ observed in each platform. However, much more work is needed to establish the theoretical framework to define ‘co-expression’ across platforms.

Availability of data and material

The data used in this manuscript is publicly available through the GEO database (accession number GSE32012) Software is freely available at <https://haim-bar.uconn.edu/software/>.

References

- Luis A Nunes Amaral, Antonio Scala, Marc Barthélemy, and H Eugene Stanley. Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21):11149–11152, 2000.
- Onureena Banerjee, Laurent El Ghaoui, and Alexandre d’Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine learning research*, 9(Mar):485–516, 2008.
- Haim Bar and Elizabeth D. Schifano. Differential variation and expression analysis. *bioRxiv*, 2018a. doi: 10.1101/276337. URL <https://www.biorxiv.org/content/early/2018/03/05/276337>.
- Haim Bar and Elizabeth D. Schifano. *DVX: an R package for Differential Variation and eXpression analysis*, 2018b. URL <https://haim-bar.uconn.edu/software/DVX/>.
- Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011.
- Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate—a practical and powerful approach to multiple testing. *Journal Of The Royal Statistical Society Series B*, 57(3):499–517, 1995.
- Tony Cai, Weidong Liu, and Yin Xia. Two-sample covariance matrix testing and support recovery in high-dimensional and sparse settings. *Journal of the American Statistical Association*, 108(501):265–277, 2013.

- Liang-Hui Chu, Corban G Rivera, Aleksander S Popel, and Joel S Bader. Constructing the angiome: a global angiogenesis protein interaction network. *Physiological genomics*, 44(19):915–924, 2012.
- Gene Ontology Consortium et al. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl 1): D258–D261, 2004.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal Of The Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- Guy Horev, Jacob Ellegood, Jason P. Lerch, Young-Eun E. Son, Lakshmi Muthuswamy, Hannes Vogel, Abba M. Krieger, Andreas Buja, R. Mark Henkelman, Michael Wigler, and Alea A. Mills. Dosage-dependent phenotypes in models of 16p11.2 lesions found in autism. *Proceedings of the National Academy of Sciences*, 108(41):17076–17081, 2011. doi: 10.1073/pnas.1114042108. URL <http://www.pnas.org/content/108/41/17076.abstract>.
- Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833): 41–42, 2001.
- Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.
- Purvesh Khatri, Marina Sirota, and Atul J Butte. Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS computational biology*, 8(2):e1002375, 2012.
- Elizaveta Levina, Adam Rothman, and Ji Zhu. Sparse estimation of large covariance matrices via a nested lasso penalty. *The Annals of Applied Statistics*, pages 245–263, 2008.
- Jun Li, Song Xi Chen, et al. Two sample tests for high-dimensional covariance matrices. *The Annals of Statistics*, 40(2):908–940, 2012.
- Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.
- Vamsi K Mootha, Cecilia M Lindgren, Karl-Fredrik Eriksson, Aravind Subramanian, Smita Sihag, Joseph Lehar, Pere Puigserver, Emma Carlsson, Martin Ridderstråle, Esa Laurila, et al. Pgc-1 α -responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature genetics*, 34(3):267–273, 2003.
- NCI and NHGRI. URL <https://cancergenome.nih.gov>. National Cancer Institute and National Human Genome Research Institute, The Cancer Genome Atlas.
- Mark EJ Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1): 016132, 2001.
- Karin Radrich, Yoshimasa Tsuruoka, Paul Dobson, Albert Gevorgyan, Neil Swainston, Gino Baart, and Jean-Marc Schwartz. Integration of metabolic databases for the reconstruction of genome-scale metabolic networks. *BMC systems biology*, 4(1):114, 2010.
- Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555, 2002.
- Adam J Rothman, Peter J Bickel, Elizaveta Levina, Ji Zhu, et al. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.
- James R Schott. A test for the equality of covariance matrices when the dimension is large relative to the sample sizes. *Computational Statistics & Data Analysis*, 51(12):6535–6542, 2007.
- Reginald D Smith. The network of collaboration among rappers and its community structure. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(02):P02006, 2006.
- Joshua M Stuart, Eran Segal, Daphne Koller, and Stuart K Kim. A gene-coexpression network for global discovery of conserved genetic modules. *science*, 302(5643):249–255, 2003.
- Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43): 15545–15550, 2005.
- Ian W Taylor, Rune Linding, David Warde-Farley, Yongmei Liu, Catia Pesquita, Daniel Faria, Shelley Bull, Tony Pawson, Quaid Morris, and Jeffrey L Wrana. Dynamic modularity in protein interaction networks predicts breast cancer outcome. *Nature biotechnology*, 27(2):199–204, 2009.

- Amy Hin Yan Tong, Guillaume Lesage, Gary D Bader, Huiming Ding, Hong Xu, Xiaofeng Xin, James Young, Gabriel F Berriz, Renee L Brost, Michael Chang, et al. Global mapping of the yeast genetic interaction network. *science*, 303(5659):808–813, 2004.
- Stephan Wuchty, Zoltán N Oltvai, and Albert-László Barabási. Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nature genetics*, 35(2):176–179, 2003.
- Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.
- Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1), 2005.
- Liqing Zhang, Layne T Watson, and Lenwood S Heath. A network of scop hidden markov models and its analysis. *BMC bioinformatics*, 12(1):191, 2011.
- Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder, John Lafferty, and Larry Wasserman. *huge: High-Dimensional Undirected Graph Estimation*, 2015. URL <https://CRAN.R-project.org/package=huge>. R package version 1.2.7.
- Lingxue Zhu, Jing Lei, Bernie Devlin, and Kathryn Roeder. Testing high-dimensional covariance matrices, with application to detecting schizophrenia risk genes. *The Annals of Applied Statistics*, 11(3):1810, 2017.

Supplementary Materials

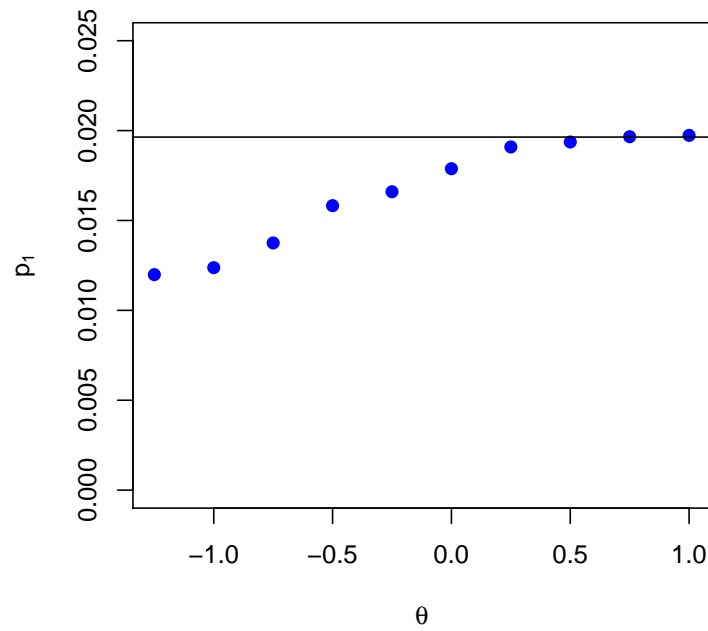


Figure S1: The average estimate of p_1 , the proportion of positively correlated pairs of genes, from 20 replications, is plotted versus θ in the *two negatively correlated blocks* configuration. The horizontal black line depicts the true value of p_1 .

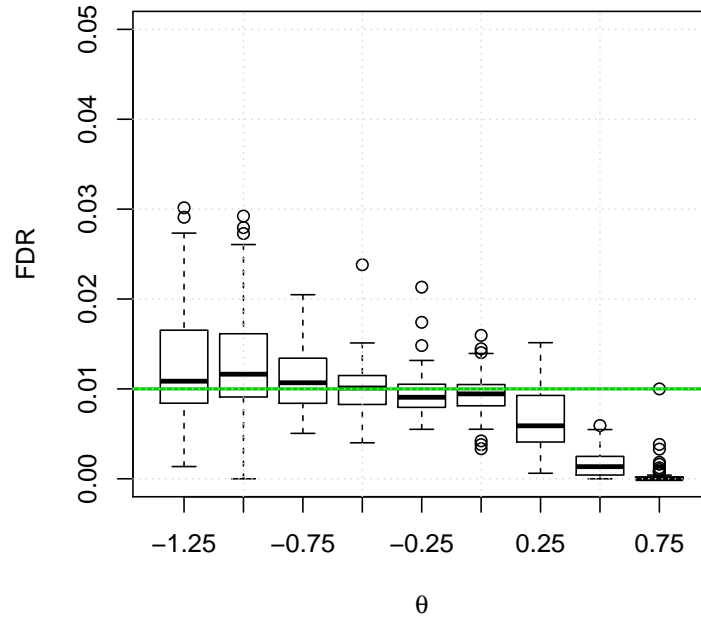


Figure S2: The observed false discovery rate in the simulations under the L_2N model (Section 3.1), as a function of the location parameter of the log-normal distributions of the non-null components. The green horizontal line shows the level (0.01) which we used to control the false discovery rate. Using the notation Section 2, we determined the thresholds, c_1 and c_2 , that correspond to this FDR level. The boxplots show the distributions across four network structures, with 20 replications in each configuration.

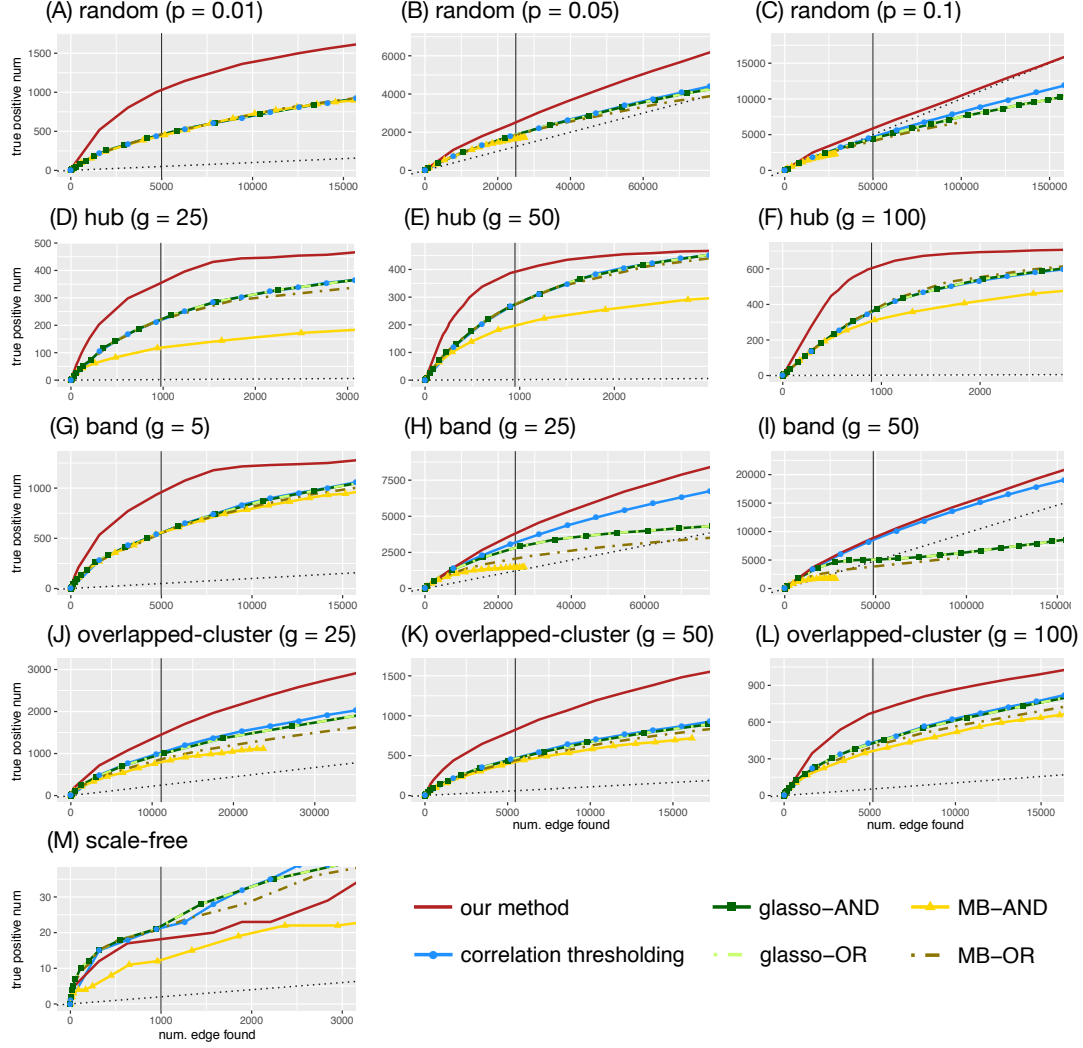


Figure S3: The numbers of true positive edges given the total number of edges identified by each method. The true matrix is obtained by applying a threshold to the true covariance matrix. The y-axis represents the number of true positive edges and the x-axis represents the total number of edges identified. The vertical line represents the number of true edges. The black dotted line is a regression line with 0 intercept and slope equal to the true sparsity, which represents the expected number of true positive edges when the edges are identified in a random manner.

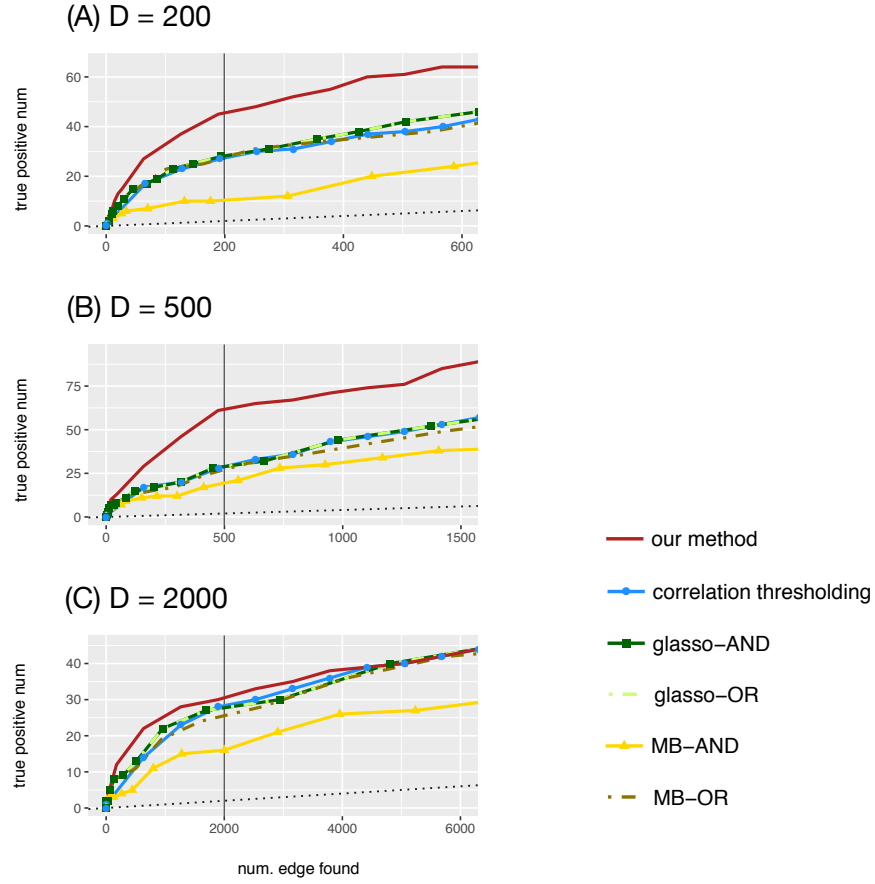


Figure S4: The numbers of true positive edges given the total number of edges using three scale-free network configurations ($D = 200, 500, 2000$). The adjacency matrix is used as the true matrix. The y-axis represents the number of true positive edges and the x-axis represents the total number of edges identified. The vertical line represents the number of true edges. The black dotted line is a regression line with 0 intercept and slope equal to the true sparsity, which represents the expected number of true positive edges when the edges are identified in a random manner.

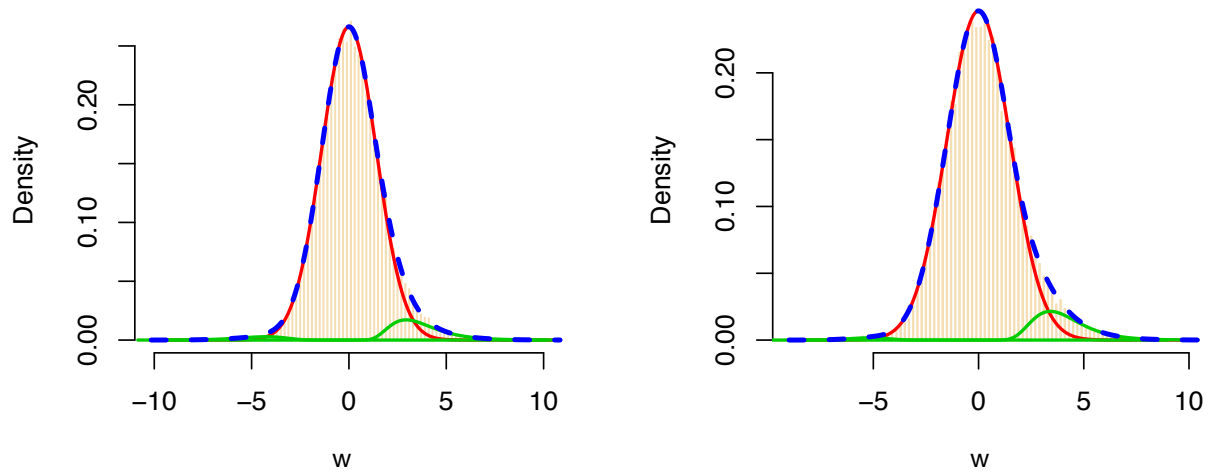


Figure S5: Goodness of fit plots of L_2N mixture model. The distributions of $w_{mn} = \text{arctanh}(r_{mn})$ for (A) wild-type and (B) duplication groups are presented. The red curve represents the null component, the green curves represent the nonnull components, and the dashed blue line represents the fitted mixture distribution.