Identification of Consistent Functional Genetic Modules

Jeffrey C. Miecznikowski[1]*, Daniel P. Gaile[1], Xiwei Chen[1], and David L. Tritchler[1,2]

[1]Department of Biostatistics, SUNY University at Buffalo, Buffalo NY 14214, USA
[2]Division of Biostatistics, University of Toronto, Toronto, ON M5T 3M7

Abstract

It is often of scientific interest to find a set of genes that may represent an independent functional module or network, such as a functional gene expression module causing a biological response, a transcription regulatory network, or a constellation of mutations jointly causing a disease. In this paper we are specifically interested in identifying modules that control a particular outcome variable such as a disease biomarker. We discuss the statistical properties that functional networks should possess and introduce the concept of network consistency which should be satisfied by real functional networks of cooperating genes, and directly use the concept in the pathway discovery method we present. Our method gives superior performance for all but the simplest functional networks.
keywords: module; network; pathway.

*Corresponding author: Jeffrey C. Miecznikowski (jcm38@buffalo.edu)

1 Introduction and objectives

It is often of scientific interest to find a set of genes that may represent an independent functional biological module or network, such as a functional gene expression module causing a biological response, a transcription regulatory network, or a constellation of mutations jointly causing a disease. This is in keeping with the *modularity* concept, where a module is a part of an organism that is integrated with respect to a certain kind of process and relatively autonomous with respects to other parts of the organism. The modularity concept has gained popularity more-or-less simultaneously in molecular biology and systems biology, developmental biology and evolutionary biology, and cognitive psychology (Wagner, Pavlicev, and Cheverud, 2007). The coordinated action of such sets of variables implies that the expressions or states of the module variables will be correlated. In addition, since the variables we are interested in are functional, their expression will correlate with the outcome variable y being studied. The identification of such modules can further the understanding of complex cellular mechanisms. In this article we will use the terms module, network and pathway interchangeably to refer to the same general concept. Although our discussion pertains to a variety of network element types, for definiteness of discussion we usually refer to the variables as genes.

An additional motivation for modeling pathways is that the power to detect important genes may be improved if the genes belong to the same module. In low signal-to-noise situations genes which are unidentifiable when evaluated individually may be discovered in a search for functional modules. The genes in a functional module may be identified by exploiting their mutual correlation and be seen as relevant in the context of a pathway or module.

We present three relevant properties a functional module should possess. The first two are the association of the module genes with each other, and the relationship of the module with the outcome variable. The third property is functional consistency, which relates the first two properties to more accurately describe real functional networks. We directly use this concept in the pathway discovery method we present. Previous approaches to pathway discovery that we discuss in this paper employ the first two properties in a sequential fashion, and do not address the third. The novel method of this paper introduces a method for identifying functional modules where co-expression and association with outcome are considered simultaneously, constrained by the consistency property.

We contrast our approach with two-step functional module-discovery approaches that incorporate information about co-expression (modularity) in one step and introduce gene-outcome association (functionality) information in a separate step. An example of a two-step method is the strategy of cluster analysis (modularity) followed by evaluating the identified clusters for the average correlation of the cluster members with the outcome variable (functionality). The functional information captured by the individual gene-outcome associations is not incorporated in the search for clusters at the first step. Much applied genomic analysis includes cluster analysis with subsequent study of the clusters. The basic approach can be implemented in a variety of ways by varying the similarity metric and the clustering method used, and has been extended and highly developed in Zhang and Horvath (2005) as part of a broad approach to network analysis they term Weighted Network Analysis (WNA).

In related work with a different objective, supervised principal components (SPC), (Bair, Hastie, Paul, and Tibshirani, 2006) finds components composed of genes which are predictive of outcome. In the first step SPC selects a set genes highly associated with outcome. Then principal components are computed for just those genes and the components are used for prediction. The number of selected genes is tuned using cross-validation to minimize the estimated out-of-sample prediction error using the principal component of the selected genes as the predictor. The components extracted by SPC are intended for prediction and not module identification. However in the specific case that a single functional module influences the outcome, the genes selected would include the module genes and the principal component would load on the module genes which are mutually correlated, so it is natural to consider the use of SPC for functional module discovery. In that case, SPC would be an example of a two-step method which considers association with outcome in the first step and then co-expression in step two, the reverse order to the clustering approach.

Specifically in Section 2 we propose and justify a statistical model for functional gene networks. In Section 3 we propose methods to estimate the model and partition the eigenvalues. In Section 4 we discuss

bagging techniques to improve the model estimation. In Section 5 we demonstrate our method and compare it with other methods via several simulations. In Section 6 we apply our methods to a simulated *E. coli* and *S. cerevisiae* datasets and an actual breast cancer dataset. We conclude the manuscript with a discussion and conclusion in Section 7.

2 Statistical model

2.1 Modularity

To represent modularity, we statistically represent gene co-expression by the correlation matrix of the gene expressions, and assume that there are groups of genes that are correlated among themselves while being uncorrelated with the other groups. Modularity implies a block structure for the appropriately ordered correlation matrix Σ. Note ordering here refers to the genes being collected or grouped into their associated pathways.

Let the vector $\underline{x} = (x_1, x_2, \ldots, x_N)$ be gene expression variables which are measured **independently** on n individuals and let the data matrix be the $n \times N$ matrix X, so each row is the gene expression measurements for an individual (i.e. a gene expression array, $\underline{x}^T$). The form for the correlation matrix Σ is given by,

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_{22} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \\ \mathbf{0} & \mathbf{0} & \cdots & \Sigma_{m-1,m-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \Sigma_{mm} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{\Delta} \end{bmatrix}, \tag{1}$$

where Σ_{jj} is a $n_j \times n_j$ within pathway correlation matrix, Δ is a $d \times d$ diagonal correlation matrix for a set D of independent genes which do not form modules, and $\Delta + \sum_{j=1}^{m} n_j = N$.

We assume that a within cluster correlation matrix Σ_{jj} arises from an independent module which can represent a biological pathway. A common simple pathway model is the single input module (SIM). We will use this model to depict common network properties and to construct simulation experiments.

2.1.1 Example: SIMs or Hub models

A SIM (Single Input Module) consists of a set genes that are controlled by a single transcription factor (Milo, Shen-Orr, Itzkovitz, Kashtan, Chklovskii, and Alon, 2002). There is considerable experimental evidence that SIMs occur frequently (Lee, Rinaldi, Robert, Odom, Bar-Joseph, Gerber, Hannett, Harbison, Thompson, Simon et al., 2002, Milo et al., 2002). For example, consider a SIM represented by the linear model for gene expression

$$x_i = \pi_i \beta x_0 + \varepsilon_i, i = 1, \ldots, t \tag{2}$$

$$x_0 = \varepsilon_0 \tag{3}$$

where $\beta > 0$, $\pi_i \in \{-1, 1\}$ and the ε_i, $i = 0, 1, \ldots, t$ are independent errors with mean 0 and variance σ_ε^2. In this framework the network consists of $t + 1$ genes, however, we often assume that the x_0 gene is latent or unobserved. The covariance of all pairs of genes in this system is nonzero. The covariation among the t observed module genes is driven by the latent unobserved *hub* x_0. The covariance of two observed (non-hub) module genes x_i and x_j denoted $cov(x_i, x_j)$ is $\pi_i \pi_j \beta^2 \sigma_\varepsilon^2$, and the covariance of x_i with the hub x_0 is $cov(x_i, x_0) = \pi_i \beta \sigma_\varepsilon^2$. The non-hub variances are $\sigma_\varepsilon^2(1 + \beta^2)$ and the correlation between observed module genes is $cor(x_i, x_j) = r_{x_i, x_j} = \frac{\pi_i \pi_j \beta^2}{1+\beta^2}$.

We model the functional aspect of the pathway by letting the hub x_0 determine an outcome variable y by the regression function

$$y = \alpha x_0 + \delta, \tag{4}$$

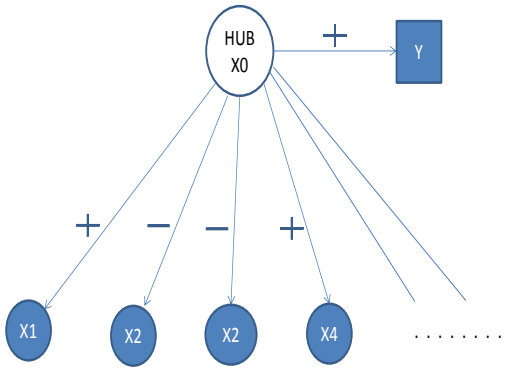

Figure 1: The graph of a SIM. The consistency property is demonstrated by tracing paths between nodes.

where without loss of generality $\alpha > 0$. Letting the variance of the error term δ in (4) be σ_δ^2, $Cov(y, x_i) = \pi_i \alpha \beta \sigma_\varepsilon^2$ for a non-hub gene x_i, the covariance of y with the hub x_0 is $\alpha\sigma_\varepsilon^2$, $Var(y) = \alpha^2 Var(x_0) + \sigma_\delta^2 = \alpha^2\sigma_\varepsilon^2 + \sigma_\delta^2$, and

$$cor(y, x_i) = \frac{\alpha}{\sqrt{\alpha^2\sigma_\varepsilon^2 + \sigma_\delta^2}} \frac{\pi_i \beta \sigma_\varepsilon^2}{\sqrt{\sigma_\varepsilon^2(1+\beta^2)}}$$

for a non-hub module gene x_i.

A model SIM is depicted in Figure 1. The unfilled circle x_0 indicates that it is unobserved, while the blue nodes are observed variables. The hub x_0 represents the latent "driver" of the pathway and the arrow from x_0 to y represents the influence of the module on y. In Figure 1, a negative sign on a link from x_0 to x_i indicates that a positive change in x_0 which is associated with an increase in y suppresses x_i, while a positive change indicates that x_i is promoted.

2.2 A statistical definition of pathway

Three natural properties of an independent functional pathway or module are:

1. *Modularity*: The variables in a specific pathway are correlated with each other and independent of the variables in other modules. That is, we adopt the model (1) for the covariance matrix.

2. *Functionality*: Each pathway variable is marginally correlated with y.

3. *Functional consistency*: The pathway variables act in concert to either increase or decrease y. Given the pattern of covariation of the pathway variables, each variable causes y to change in the same direction - the pathway variables influence y in a consistent fashion.

Consistency requires a pathway to be a set of cooperating variables. For example suppose the expression of a certain module gene promotes an increase in y while another gene in that module suppresses y. If the module is activated to increase y the promoter expression must increase while the suppressor decreases. Thus promoters and suppressors will be negatively correlated. The concept of consistency of functional effects is formalized in the following definition.

Definition: The effect of two variables x_i and x_j on y is functionally consistent when

i) if $Cor(x_i, x_j)$ is positive, then $Cor(x_i, y)$ and $Cor(x_j, y)$ have the same sign, and

ii) if $Cor(x_i, x_j)$ is negative, then $Cor(x_i, y)$ and $Cor(x_j, y)$ have opposite signs.

Tracing paths in Figure 1 shows that a SIM has the consistency property. Path tracing rules in Wright (1934) imply that the correlation r between variables is calculated by multiplying the correlation assigned to the links on the paths joining them. For example, in Figure 1, $r_{x_1,x_2} = r_{x_1,x_0} r_{x_0,x_2}$ is negative. Likewise, $r_{x_1,y} = r_{x_1,x_0} r_{x_0,y}$ is positive and $r_{x_2,y} = r_{x_2,x_0} r_{x_0,y}$ is negative so that consistency is satisfied. We can verify that consistency holds throughout the SIM. The concept of functional consistency is related to the notion of balance in a signed graph, which has been applied to the sociometric structure of groups in social psychology (Harary et al., 1953). The definition leads directly to the following lemma,

Lemma 1. *For any $i \neq j$ in a pathway, the number of negative elements in the set $\{Cor(x_i, x_j)$, $Cor(x_i, y)$, $Cor(x_j, y)\}$ is even.*

Proof. Assume $Cor(x_i, x_j)$ is positive. Then consistent effect implies that either both $Cor(x_i, y)$ and $Cor(x_j, y)$ are positive or both are negative so the number of negative signs is 0 or 2. Alternatively, if $Cor(x_i, x_j)$ is negative consistent effect implies that exactly one of $Cor(x_i, y)$ and $Cor(x_i, y)$ is negative so the number of negative signs is 2. □

Suppose that a potential pathway is being investigated for functionality. For example, after a cluster analysis the potential causal effect of a cluster on y is a typical question. The following theorem leads to a diagnostic measuring the conformity of that set of variables with the concept of consistent effect in Lemma 1,

Theorem 1. *Let X^+ be the set $\{x_i : Cor(x_i, y) > 0\}$ and X^- be the set $\{x_i : Cor(x_i, y) < 0\}$. Then consistent effect implies that*

$Cor(x_i, x_j) > 0$ *when* $x_i \in X^+$ *and* $x_j \in X^+$,

$Cor(x_i, x_j) > 0$ *when* $x_i \in X^-$ *and* $x_j \in X^-$,

$Cor(x_i, x_j) < 0$ *when* $x_i \in X^+$ *and* $x_j \in X^-$ *or* $x_i \in X^-$ *and* $x_j \in X^+$.

Proof. The sign of $Cor(x_i, x_j)$ is determined according to the Lemma. □

Theorem 1 suggests a diagnostic to evaluate if a set of genes such as produced by a cluster analysis is varying consistently with respect to its association with y. Based on the association with y form the sets X^+ and X^- as in Theorem 1 and let $X^{++} = \{(i,j); i < j, x_i \in X^+, x_j \in X^+\}$, $X^{--} = \{(i,j); i < j, x_i \in X^-, x_j \in X^-\}$, $X^{+-} = \{(i,j); x_i \in X^+, x_j \in X^-\}$ and define the Consistency C to be

$$C = \frac{\sum_{(i,j)\in X^{++}} Cor(x_i, x_j) + \sum_{(i,j)\in X^{--}} Cor(x_i, x_j) - \sum_{(i,j)\in X^{+-}} Cor(x_i, x_j)}{\sum_{i<j} |Cov(x_i, x_j)|}, \tag{5}$$

where C will be 1 when the $x's$ have perfectly consistent effects on y and will be -1 when consistency is violated to the maximum degree.

The main result of this paper is a pathway identification method which explicitly uses consistency to determine the set of genes satisfying our understanding of how a pathway should behave. To this end, we define a matrix W which captures our three criteria for a pathway, including consistency of effect. Define the *cofunction matrix* W as

$$W = diag(Cor(\underline{x}, y)) \Sigma \, diag(Cor(\underline{x}, y)), \tag{6}$$

where we add *diag* to denote that $diag(Cor(\underline{x}, y))$ is a diagonal matrix with element i on the diagonal given by $Cor(x_i, y)$. Note that elements of W are such that $w_{ij} = Cor(x_i, y) Cor(x_j, y) Cor(x_i, x_j)$. When both

x_i and x_j are associated with y, $Cor(x_i, y)Cor(x_j, y)$ will be of large magnitude - if each of them is in some functional pathway it will capture that. Likewise $Cor(x_i, x_j)$ will have large magnitude if x_i and x_j are in the *same* pathway. Then w_{ij} will be large when x_i and x_j are in the same functional pathway. To depict this, note that W can be written as

$$W = \Sigma \cdot (Cor(\underline{x}, y)Cor(\underline{x}, y)^T), \tag{7}$$

where the operator $\cdot$ denotes elementwise multiplication. The left matrix in (7) contains the information on the modular structure, while the right matrix in (7) is positive when the variables corresponding to that element are both associated with y. Figure 2 graphically depicts the formation of W for a fictitious set of simulated modules.

Clearly W captures Property 1 since zeroes in Σ force zeros in W. By Property 2, modules in Σ which are not functional vanish from W. The third property implies that W is composed of positive elements, resembling a similarity matrix. To see this, recall that w_{ij} is of the form $Cor(x_i, x_j)Cor(x_i, y)Cor(x_j, y)$. By the lemma the number of negative factors is even so in a consistent pathway w_{ij} is non-negative. Thus to enforce consistency we transform negative elements of W to be zero.

By modularity W will inherit the block diagonal structure of Σ. In this article we assume that a single module is governing the variable y in the experiment. To discover the blocks (modules) of W we will estimate its eigenstructure.

2.3 Eigenanalysis

It is clear that if W_{ii} is a block of W which when considered as a separate matrix has eigenvalue λ_i and eigenvector $\mathbf{v}_i$, the augmented vector

$$(0, 0, 0, \cdots, 0, \mathbf{v}_i, 0, \cdots, 0)^T \tag{8}$$

with $\mathbf{v}_i$ in the position corresponding to block i, is an eigenvector of W with eigenvalue λ_i and so "picks out" the pathway when searching for the non-zero elements in the eigenvector.

Our model treats a module or pathway as a block of W with positive elements. A square block of dimension m which is approximated by μE where E is the $m \times m$ matrix of 1's (so the block is approximately homogeneous) and $\mu > 0$ has one eigenvalue equal to μm and the rest equal to 0. That is, $\mu E \mathbf{1}_m \frac{1}{\sqrt{m}} = \mu m \mathbf{1}_m \frac{1}{\sqrt{m}}$, so the eigenvector corresponding to μm is $\mathbf{1}_m \frac{1}{m}$ where $\mathbf{1}_m$ is the m-vector whose elements are all 1. The remaining eigenvectors must be orthogonal to $\mathbf{1}_m$ and thus have eigenvalue 0 since they satisfy $E\mathbf{x} = 0\mathbf{x}$. A corresponding result for random matrices composed of independent elements with mean μ is proven in Juhasz (1989) thus showing the above statement is approximately true in the random case.

The above considerations imply that the eigenvectors corresponding to the largest eigenvalues of W will identify the blocks of W and thus the pathways. Since the eigenvalue is a function of block size, larger modules will tend to have larger eigenvalues, as will modules with numerically larger elements in W (a function of the covariance of the genes with each other and with outcome). Thus the eigenanalysis will identify eigenvalues in decreasing order, so higher cardinality modules with stronger correlations and effect sizes will be found first.

We modify W in order to minimize the effect of functional genes which are not in a pathway and which operate independently of the other genes studied. Such a gene x_i may be considered to be a block of size 1 with $w_{ij} = 1 \cdot cor(x_i, y)^2$. The eigenvalue corresponding to this block is $cor(x_i, y)^2$ and the eigenvector will have 1 in the i^{th} position and zeros elsewhere. We do not want to select these elements so we alter W to give them eigenvalue zero by using the modified matrix $W_0 = W - diag(W)$, a common practice when defining an association matrix. Note $diag(W)$ is a matrix with zeroes on the off diagonal and the diagonal elements of W on its diagonal. The eigenvalue of a valid size m functional pathway will change only slightly by a factor of $\frac{m-1}{m}$ when using W_0 compared to W.

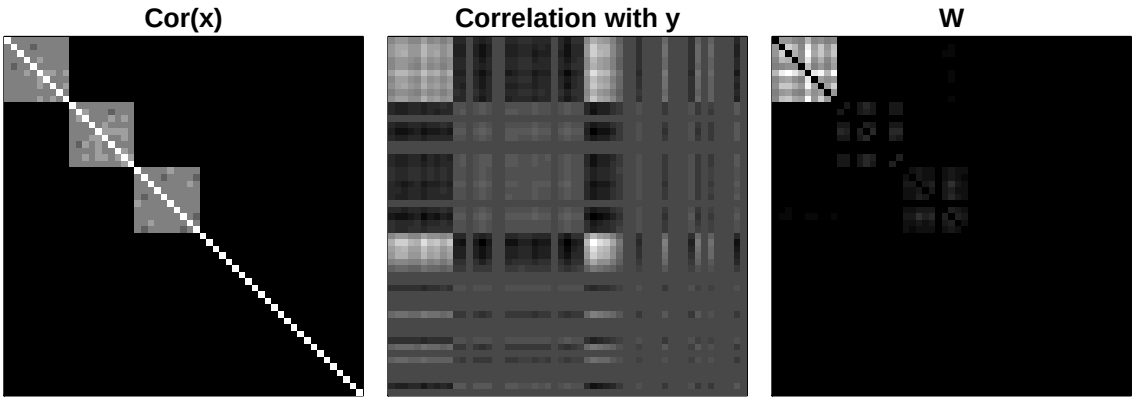

Figure 2: The formation of $W = \Sigma \cdot (Cor(\underline{x}, y)Cor(\underline{x}, y)^T)$ for a fictitious simulated module. The leftmost matrix is Σ, the middle matrix is $Cor(\underline{x}, y)Cor(\underline{x}, y)^T$, and the rightmost matrix is their element-wise product W.

3 Estimation and partitioning the pathway eigenvector

To implement our method we substitute sample estimates (Pearson correlation) for the correlations forming W in (7). Theoretically the non-pathway elements of the first eigenvector of W_0 will be zero, but of course this does not occur with real data. The eigenvector will be a mixture of two components: the elements corresponding to the pathway genes, and the remaining elements corresponding to non-pathway genes which we assume to be of smaller magnitude. One approach to partitioning the vector is choosing a mixture analysis method to partition the vector. Another approach would be to use a ι_1-regularization method to obtain a sparse eigenvector and identify the pathway as the non-zero elements of the sparse vector. Future work will focus on studying possible partitioning techniques for our method. In this paper, we apply a very simple method to partition the eigenvector. We use k-means clustering on the first eigenvector to find two clusters of the vector elements. The cluster with the larger-magnitude elements is then taken to be the module. This simple method has worked well in practice, does not depend on distributional assumptions, and does not require the specification of a tuning parameter.

4 Bagging

Bootstrap aggregation or "bagging" in Breiman (1996) is a simple and very general approach to improve upon an unstable estimator for a given set of data. Bagged eigenvectors of W_0 are obtained from an eigen-decomposition based on re-sampling entire rows from the original data with replacement so that the bootstrap sample is consistent with the original dimensions of the data matrix. The primary advantages of bagged eigenvectors are smaller variance (large effective sample size) and higher accuracy as a point estimate, at the cost of slower computation. In short, bagging is essentially a variance reduction method (Breiman, 1996). Another interpretation of the bagged estimate is as an approximate Bayesian posterior mean estimate (Hastie, Tibshirani, Friedman, Hastie, Friedman, and Tibshirani, 2009). It allows us to construct a small-sample estimator of the first eigenvector for the theoretical W_0.

Two major shortcomings of the eigen-decomposition based on bootstrap samples are: (1) reflection, which is the arbitrary change in the sign of the eigenvectors; and (2) reordering where two or more eigenvalues have very similar magnitude (Jackson, 1995). In the latter case, eigenvectors obtained from bootstrap samples may come out altered in their order relative to that found from the original sample. Under either condition, it may lead to the false acceptance of the null hypothesis that eigen-vector coefficients are not significantly different from zero. In order to address these drawbacks, we propose the following algorithm. Note that we focus on the rank-1 approximation to the modified matrix W_0 and the corresponding first eigenvector of the modified matrix W_0 in order to identify one pathway.

1. Generate a bootstrap sample $(X, y)^{*b}$ with replacement of samples from the original data and calculate the matrix $W^{*b} = diag(Cor(\underline{x}^{*b}, y^{*b}))\Sigma^{*b}\, diag(Cor(\underline{x}^{*b}, y^{*b}))$. Set the negative elements of W^{*b} to 0 and then obtain the modified matrix $W_0^{*b} = W^{*b} - diag(W^{*b})$. Repeat this process for each $b = 0, 1, \ldots, B$ independently (e.g. with $B = 1000$).

2. For each matrix W_0^{*b},

   (a) Calculate the correlation r_j^{*b} between the first eigenvector $\mathbf{v}$ of the W_0 matrix based on the original data and the eigenvectors $\mathbf{v}_j^{*b}$ of W_0^{*b} based on the bootstrapped data, i.e., $r_j^{*b} = cor(\mathbf{v}, \mathbf{v}_j^{*b})$, $j = 0, 1, \ldots, N$.

   (b) Find $\xi^{*b} = argmax_j |r_j^{*b}|$. This procedure is equivalent to performing orthogonal rotations and correcting for reversals in the axis ordering (Milan and Whittaker, 1995).

   (c) Inspect the sign of $r_{\xi^{*b}}^{*b}$. A negative correlation indicates a reflection and the bagged eigenvector should be converted by multiplying the elements by -1. The first eigenvector based on the bootstrap sample is $\mathbf{v}_{\xi^{*b}}^{*b} sign(r_{\xi^{*b}}^{*b})$.

3. Aggregate the bootstrap estimates by computing the bootstrap mean

$$\bar{\mathbf{v}} = (1/B)\sum_{b=1}^{B} \mathbf{v}^{*b}_{\xi^{*b}} sign(r^{*b}_{\xi^{*b}}). \tag{9}$$

4. Run k-means on $\bar{\mathbf{v}}$ assuming 2 clusters ($k = 2$).

5 Simulations

We simulated an instance of (1) with 5 modules of 20 genes each and a set of 300 independent genes in Δ. The 5 modules are SIM's, described by (2) and Figure 1. Only one of the modules is functional as per (4), the other four are uncorrelated with the outcome variable y. We set the model parameters to produce a specified signal-to-noise ratio (SNR) and intra-modular correlation r_m. We defined SNR to be the ratio of $\alpha\sigma_\varepsilon$ to σ_δ, that is, the square root of the ratio of the variance in y accounted for by the hub to the variance of the error component.

We compared our methods including clustering just the observed eigenvector (W.e) and clustering the bagged eigenvector (W.be), SPC, and three cluster analysis based methods. For supervised principal components we used the package `superpc` in R version 3.0.2 to calculate the first supervised principal component (Bair and Tibshirani, 2012). The clustering methods were k-means clustering using the data matrix X (K6), the partitioning around medoids algorithm (WGC6.r) using the absolute value of correlations of X (Pam.cor) and partitioning around medoids (PAM) using the correlations of X to the sixth power (WGC6.r6), which tends to differentially diminish small correlation coefficients with small magnitude. Note WGC6.r and WGC6.r6 are variants of Weighted Network Analysis (WNA) described in Horvath (2011). The clustering methods were set to obtain six clusters, the true number of clusters. However, in practice identifying the true number of clusters is difficult. The cluster analysis based methods computed the average significance for predicting y from the genes in each cluster (the module significance) and the functional module was taken to be the cluster with the highest module significance.

All methods are evaluated and compared in scenarios of coefficients with the same sign or random signs. In the simple case of coefficients with the same sign, we let $\pi_i = 1$ in (2) for all module genes, so that all intramodular correlations were the same sign. For a more general network of coefficients with random signs, we randomly selected each π from the set $\{-1, 1\}$ so that a module could consist of a mix of promoters and suppressors.

For SNRs of .3 and .5 we calculated the true positive rate (TPR) and the false discovery rate (FDR) for each method, for a range of intramodule absolute correlation (Rm) levels from .1 to .7. TPR was calculated as the proportion of the true module genes selected by the method. FDR is the proportion of selected genes which actually are module genes. 200 Monte Carlo simulated samples were generated for each model for sample sizes of 100 and 300 and the averages of TPR and FDR over the simulations were calculated.

Figures 3 through 6 were generated using coefficients with the same sign. Figures 3 through 4 show results for case of constant coefficient sign and sample size of 100. In that setting, K6 outperformed all the other methods. When both TPR and FDR are taken into account the proposed methods W.e and W.be are preferred to all but K6. Note SPC performs the worst, which is not surprising since it is optimized for prediction and was not intended for module discovery. With the larger SNR value of .5, FDRs of the proposed methods and K6 start to stabilize around values below 0.05 at the intramodule absolute correlation level of 0.4. The remaining methods have unacceptable FDR. The same pattern holds for sample size 300 (Figure 5 and Figure 6). At that sample size the proposed methods and K6 have FDRs below 0.05 when the intramodule absolute correlation level is 0.3 or greater for SNR is .3 (Figure 5) and for SNR= .5 the FDR is close to 0 when the intramodule absolute correlation level is 0.2 or greater (Figure 6).

When random signs are generated (Figures 7 through Figures 10) the proposed methods outperform the remaining methods including K6. Although WGC6.r has higher TPR for Rm> .2 it has a very high FDR and is not a competitor. K6 does very poorly in this scenario. We attribute the superiority of the proposed

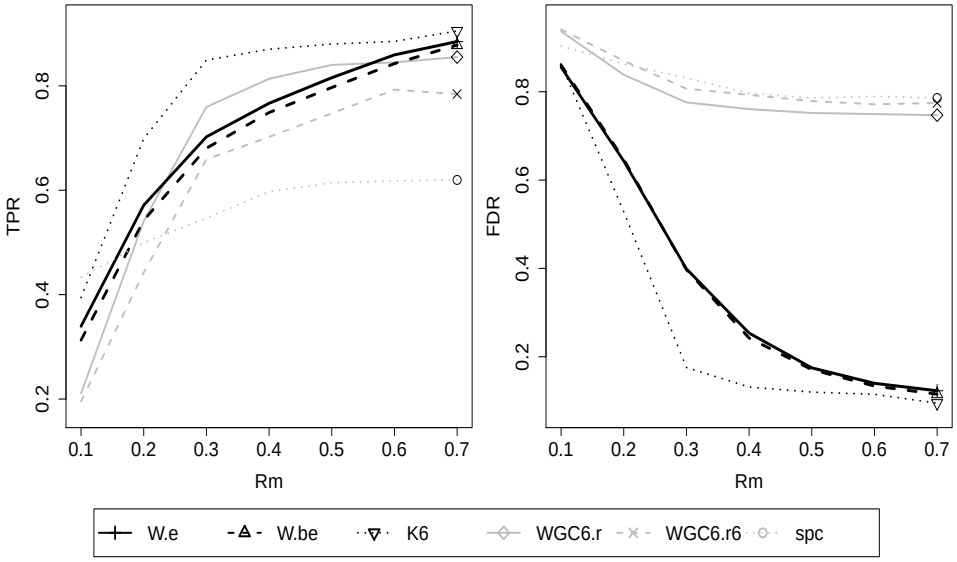

Figure 3: True positive rate and false discovery from simulation study with SNR of .3, a sample size of 100 and coefficients with the same sign. Rm is the magnitude of the correlation between module genes.

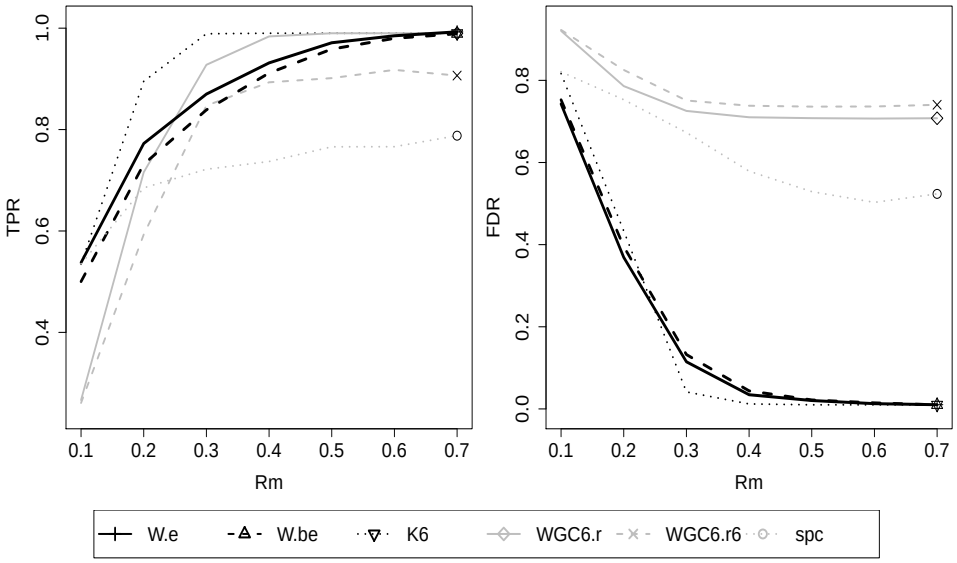

Figure 4: True positive rate and false discovery from simulation study with SNR of .5, a sample size of 100 and coefficients with the same sign. W.e and W.be are our basic and bootstrap pathway discovery method. K6, WGC6.r, and WGC6.r6 are variants of Weighted Network Analysis (WNA), using respectively k-means for 6 clusters, PAM for 6 clusters using absolute correlation, and PAM for 6 clusters using correlation to the sixth power. SPC is Sparse Principal Components.

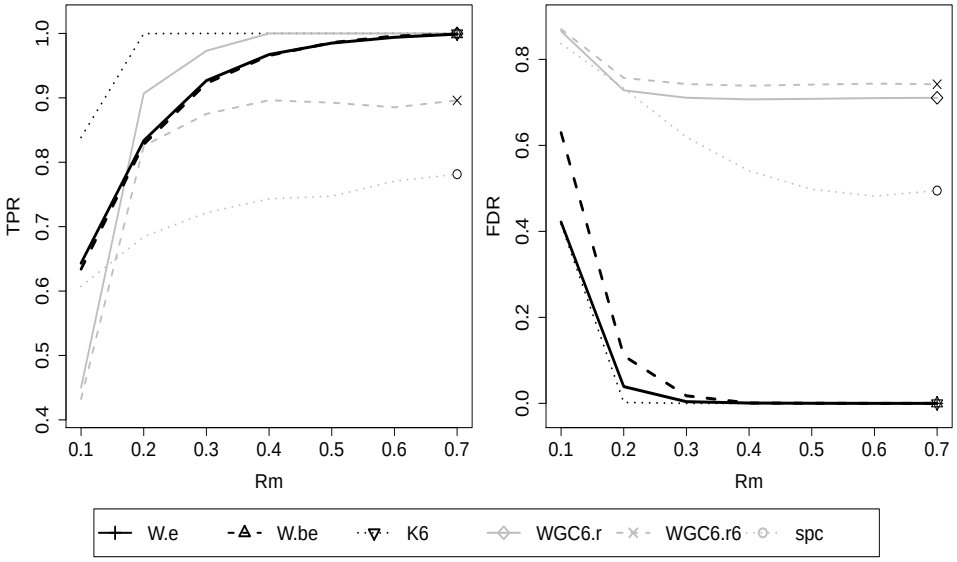

Figure 5: True positive rate and false discovery from simulation study with the SNR of .3, a sample size of 300 and coefficients with the same sign.

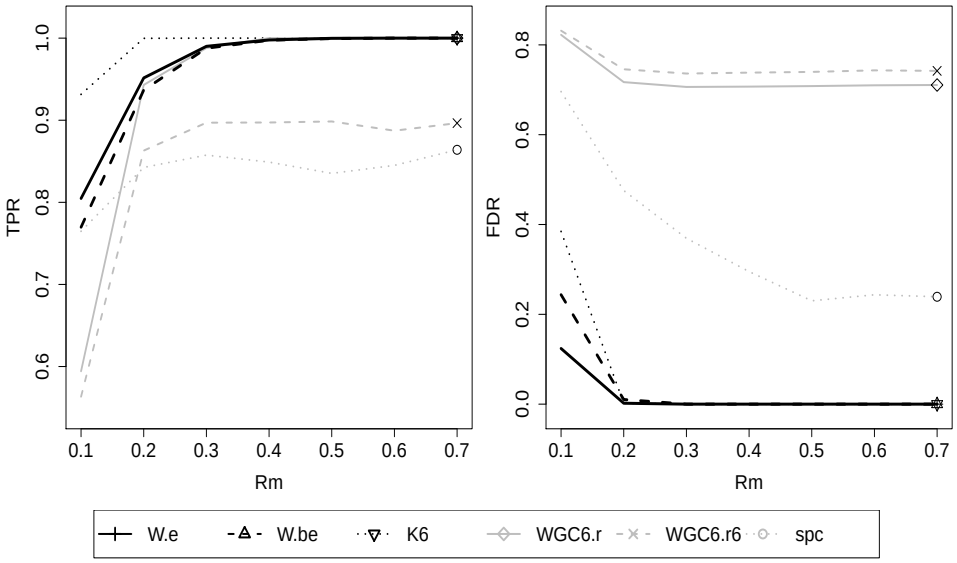

Figure 6: True positive rate and false discovery from simulation study with the SNR of .5, a sample size of 300 and coefficients with the same sign. W.e and W.be are our basic and bootstrap pathway discovery method. K6, WGC6.r, and WGC6.r6 are variants of Weighted Network Analysis (WNA), using respectively k-means for 6 clusters, PAM for 6 clusters using absolute correlation, and PAM for 6 clusters using correlation to the sixth power. SPC is Sparse Principal Components.

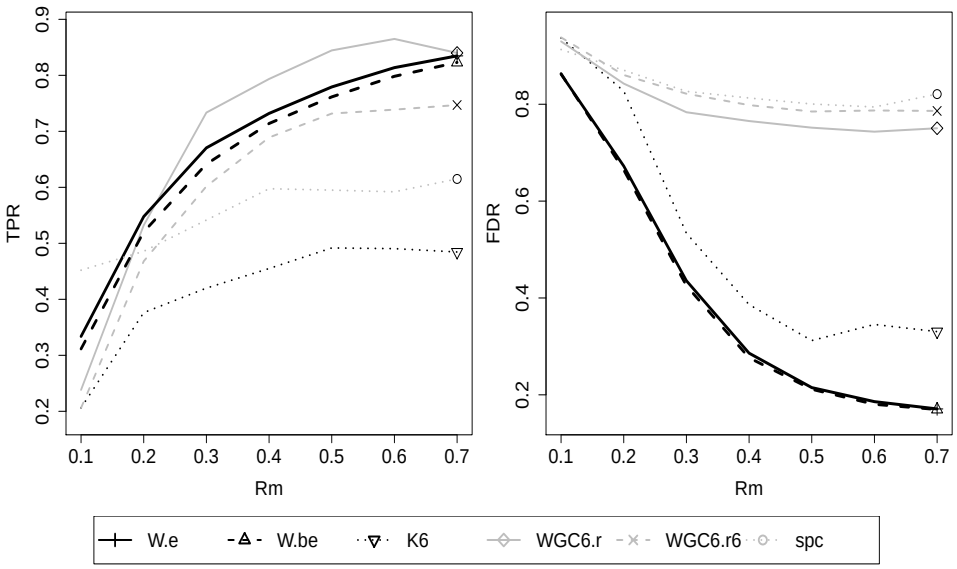

Figure 7: True positive rate and false discovery from simulation study with the SNR of .3, a sample size of 100 and coefficients with random signs. W.e and W.be are our basic and bootstrap pathway discovery method. K6, WGC6.r, and WGC6.r6 are variants of Weighted Network Analysis (WNA), using respectively k-means for 6 clusters, PAM for 6 clusters using absolute correlation, and PAM for 6 clusters using correlation to the sixth power. SPC is Sparse Principal Components.

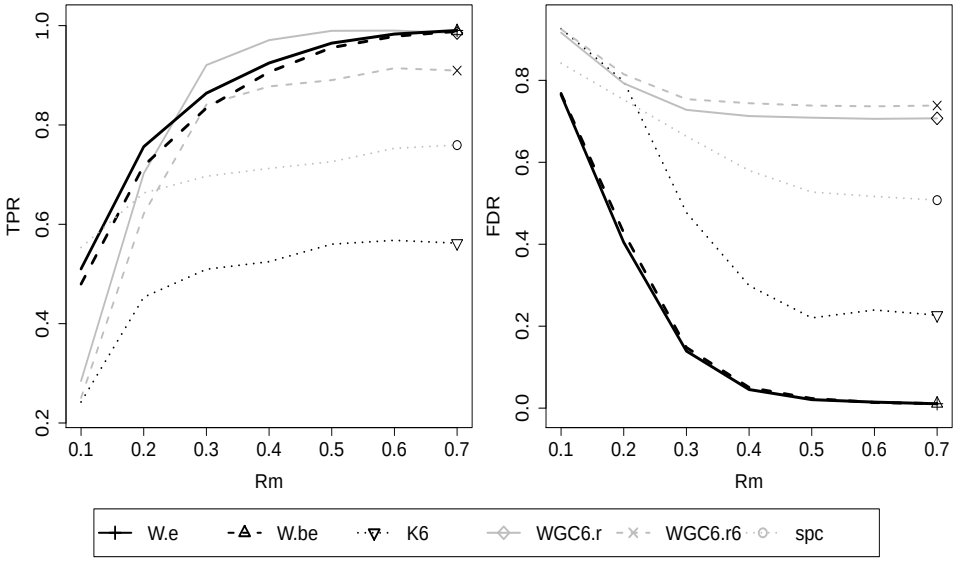

Figure 8: True positive rate and false discovery from simulation study with the SNR of .5, a sample size of 100 and coefficients with random signs. W.e and W.be are our basic and bootstrap pathway discovery method. K6, WGC6.r, and WGC6.r6 are variants of Weighted Network Analysis (WNA), using respectively k-means for 6 clusters, PAM for 6 clusters using absolute correlation, and PAM for 6 clusters using correlation to the sixth power. SPC is Sparse Principal Components.

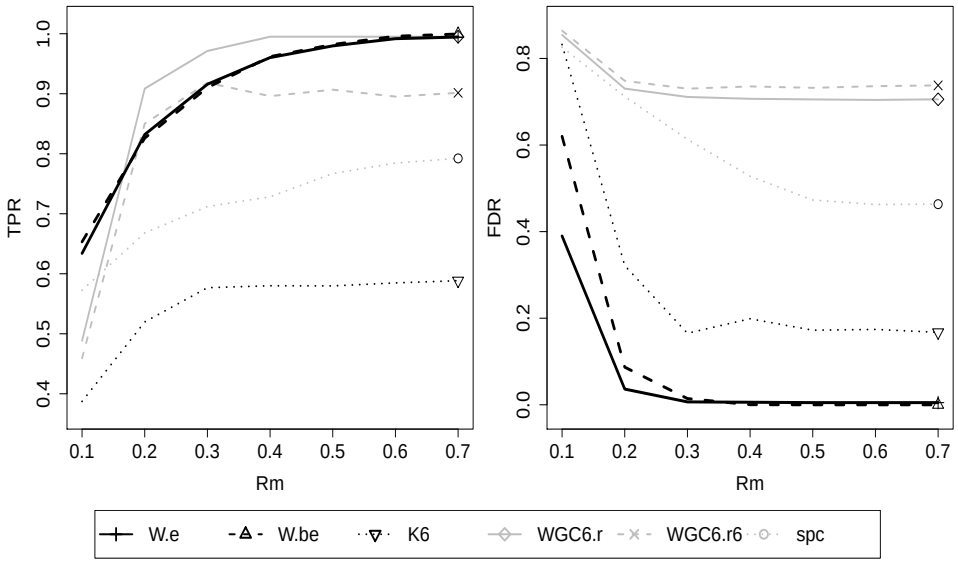

Figure 9: True positive rate and false discovery from simulation study with the SNR of .3, a sample size of 300 and coefficients with random signs. W.e and W.be are our basic and bootstrap pathway discovery method. K6, WGC6.r, and WGC6.r6 are variants of Weighted Network Analysis (WNA), using respectively k-means for 6 clusters, PAM for 6 clusters using absolute correlation, and PAM for 6 clusters using correlation to the sixth power. SPC is Sparse Principal Components.

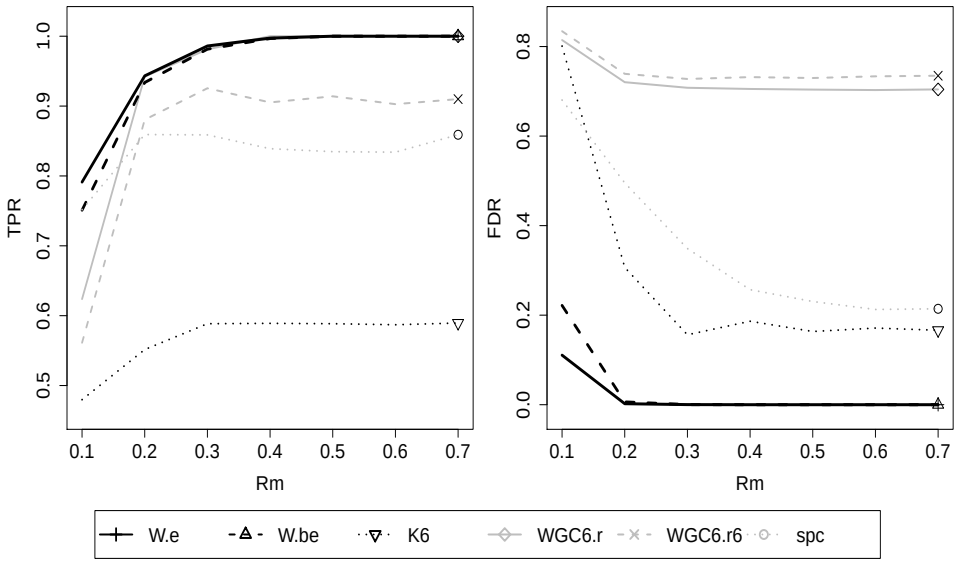

Figure 10: True positive rate and false discovery from simulation study with the SNR of .5, a sample size of 300 and coefficients with random signs. W.e and W.be are our basic and bootstrap pathway discovery method. K6, WGC6.r, and WGC6.r6 are variants of Weighted Network Analysis (WNA), using respectively k-means for 6 clusters, Pam for 6 clusters using absolute correlation, and Pam for 6 clusters using correlation to the sixth power. SPC is Sparse Principal Components.

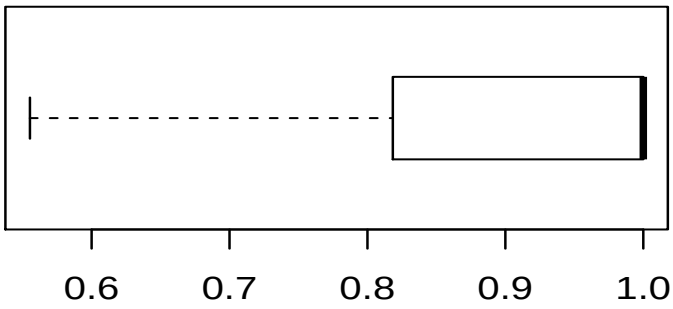

Figure 11: Distribution of proportion of dominant signs from modules found by k-means clustering.

Methods	Rm=0.1	Rm=0.2	Rm=0.3	Rm=0.4	Rm=0.5	Rm=0.6	Rm=0.7
K6	0.2339	0.3564	0.6479	0.8011	0.8602	0.8796	0.9064
WGC6.r	0.0978	0.1698	0.2613	0.3444	0.4121	0.4708	0.5206
WGC6.r6	0.0981	0.1580	0.2356	0.3085	0.3740	0.4381	0.4843
spc	0.2818	0.3291	0.3725	0.4068	0.4200	0.4464	0.4555
W.e	0.3096	0.4191	0.5585	0.6630	0.7081	0.7286	0.7370
W.be	0.3412	0.4213	0.5659	0.6795	0.7207	0.7349	0.7418

Table 1: Average consistency of modules detected.

methods under varying directions of gene-gene association to the use of consistency in defining groups of genes. To investigate this, Table 1 shows the average functional consistency (5) over 100 simulations using the parameters of Figure 7, that is, SNR= .3, a sample size of 100 and coefficients with random signs. We see that the proposed methods find more consistent modules except for the case of k-means which is most consistent. This seems surprising since k-means performed poorly in this situation (Figure 7). Apparently, k-means clustering tends to produce clusters which tend to have either all positive or all negative gene-gene correlation. This is shown by plotting the distribution of the proportion of gene pairs with associations which are the dominant sign (positive or negative) for clusters picked up by k-means over 200 simulations with Rm= $0.3, SNR = 0.3$, and random sign (Figure 11). We see that most of the genes detected have correlations of the same sign. Thus k-means will omit genes with differing correlation and can only find subsets of the functional module, which results in the poor TPR shown in Figure 7.

To summarize, the proposed methods perform relatively well for all conditions and are clearly superior in the case of random signs, which represents a realistic biological network composed of both promoters and suppressors.

Measures	Methods	SNR = 0.2	SNR = 0.3	SNR = 0.4	SNR = 0.5	SNR = 0.6
TPR	K-means	0.5092	0.5190	0.5290	0.5340	0.5423
	WGC	0.7998	0.8026	0.8041	0.8044	0.8046
	WGC.r6	0.8055	0.8104	0.8118	0.8120	0.8118
	SPC	0.4086	0.4836	0.5341	0.5568	0.5517
	W.e	0.6222	0.6214	0.6219	0.6222	0.6223
	W.be	0.6149	0.6204	0.6218	0.6221	0.6223
FDR	K-means	0.8920	0.8894	0.8875	0.8869	0.8866
	WGC	0.8579	0.8574	0.8572	0.8572	0.8571
	WGC.r6	0.8585	0.8576	0.8573	0.8572	0.8572
	SPC	0.7841	0.7161	0.6629	0.6359	0.6131
	W.e	0.0787	0.0199	0.0041	0.0020	0.0000
	W.be	0.0593	0.0179	0.0031	0.0020	0.0000

Table 2: *E. coli* example

6 Examples

In this section we explore several examples. In the first example, we apply our methods to synthetic (simulated) datasets designed to simulate regulatory networks specific to specific organisms (e.g. *S. Cerevisiae* and *E. coli*). In the second example, we apply our methods to a human breast cancer gene expression microarray dataset. In this way, we "connect the dots" between the data in Section 5 involving purely theoretical modules, to data simulating naturally occurring modules within an organism (e.g. yeast), to data where it is unknown what functional modules may exist (breast cancer).

The modular network models we have simulated certainly do not match the complexity of real microarray data. However, with real data the true underlying model is unknown so it is impossible to know which genes selected are true positives or misclassified irrelevant genes. To satisfy the criteria of realism and known properties, we generated datasets using SynTReN (Van den Bulcke, Van Leemput, Naudts, van Remortel, Ma, Verschoren, De Moor, and Marchal, 2006) or **Syn**thetic **T**ranscriptional **Re**gulatory **N**etworks, a generator of synthetic gene expression data. This approach allows a quantitative assessment of the accuracy of the methods applied. The SynTReN generator generates a network topology by selecting subnetworks from the well characterized *E. coli* or *S. cerevisiae* regulatory networks. Then transition functions and their parameters are assigned to the edges in the network. Eventually, mRNA expression levels for the genes in the network are obtained by simulating equations based on Michaelis-Menten and Hill kinetics under different conditions (Chou, 1976). After the addition of noise, microarray gene expression measurements are produced.

We produced two synthetic expression datasets, one based on the *E.coli* network topology and one for *S.cerevisiae*. In each dataset there were 30 functional network genes and 300 background genes. The functional network was perturbed though an exogenous gene x_0. The 300 background genes have an underlying network structure, but are not perturbed and so propagate only error. This is a more realistic model for inactive genes than in our simulations. We used the cluster addition option of SynTReN and set all parameters to their default values.

To introduce functionality we simulated y according to (4). We defined SNR to be the ratio of $\alpha\sigma_\varepsilon$ to σ_δ. For SNRs ranging from 0.2 to 0.6, we calculated the average TPR and FDR for the same pathway detection methods as section 5 over 800 Monte Carlo simulations with a sample size of 400. Tables 2 and 3 show the results for the *E. coli* and *S. cerevisiae* organism, respectively. All of the methods except the proposed methods W.e and W.be have unacceptable FDR. Only WGC and WGC.6 have higher TPR but their FDR is over 80%. The proposed methods alone have FDRs in a range that allows them to be useful in practice. The results mirror the simulation in Section 5 with random signs, which suggests that the random sign model more accurately reflects real biological networks.

Measures	Methods	SNR = 0.2	SNR = 0.3	SNR = 0.4	SNR = 0.5	SNR = 0.6
TPR	K-means	0.4799	0.4875	0.4881	0.4885	0.4946
	WGC	0.7886	0.7941	0.7930	0.7929	0.7914
	WGC.r6	0.7889	0.7942	0.7934	0.7929	0.7929
	SPC	0.3895	0.4585	0.4548	0.4465	0.4448
	W.e	0.6239	0.6100	0.6071	0.6067	0.6067
	W.be	0.6069	0.6066	0.6067	0.6067	0.6067
FDR	K-means	0.8836	0.8824	0.8819	0.8804	0.8804
	WGC	0.8601	0.8590	0.8592	0.8593	0.8595
	WGC.r6	0.8606	0.8596	0.8597	0.8597	0.8597
	SPC	0.8045	0.6990	0.6023	0.5314	0.4977
	W.e	0.0793	0.0218	0.0040	0.0000	0.0000
	W.be	0.0254	0.0035	0.0000	0.0000	0.0000

Table 3: *S. cerevisiae* example.

We also explore the ability to implement our techniques on a real dataset. We examine the "Desmedt" dataset - a breast cancer microarray dataset described in Desmedt, Piette, Loi, Wang, Lallemand, Haibe-Kains, Viale, Delorenzi, Zhang, d'Assignies et al. (2007) and explored in Miecznikowski, Wang, Liu, Sucheston, and Gold (2010). In this dataset we have 198 tumor samples from breast cancer patients assayed on the Affymetrix U133A microarrays. The dataset was pre-processed using the RMA algorithm in the "affy" package within the Bioconductor suite of `R` packages (Gautier, Cope, Bolstad, and Irizarry, 2004, Gentleman, Carey, Bates, and others, 2004). There are 22,283 probes/genes in the dataset and for each patient we have censored survival times measured in days. Using the method described in Klein, Gerster, Andersen, Tarima, and Perme (2008) and the `R` package (Maja Pohar Perme and Gerster, 2012) we compute pseudo-values from the censored survival time. In short, these pseudo values are the survival probabilities for each individual at the pre-specified time point (in our case, the median survival time). These pseudo-values act as the outcome variable y while the microarray gene expression data is X. Applying the bootstrapped version of our method to this data with 200 bootstrap replications in the bagging step yields a functional pathway consisting of approximately 1200 genes.

The heatmap for the top 30 genes (out of ~ 1200) is shown in Figure 12. The top genes were according to the size (in absolute value) of their eigenvalue. As expected, it appears that each gene is functionally correlated with each other (either positively or negatively) as well as with the outcome.

7 Discussion and Conclusion

Our pathway discovery method is easily implemented, only requiring routine calculations, eigenanalysis, and k-means clustering with 2 clusters. The multiplication as shown in (6) can be memory intensive and computationally intensive for large matrices. However, note the multiplication in (6) involves three matrices where the left most and right most are diagonal matrices. The diagonal matrix on the left multiplies each row in the middle matrix by its corresponding diagonal element and the diagonal matrix on the right multiplies each column in the middle matrix by the corresponding diagonal element. This operation (and thus the matrix multiplication) can be quickly and efficiently implemented in `R` by using the `sweep` command. Also note, the eigenanalysis in `R` can be simply implemented using the Csardi and Nepusz (2006) package which is an interface to the ARPACK library for calculating eigenvectors. Note, in implementing our approach to assess eigenvalue reflection (Step 2(a) of Section 4), we propose calculating N eigenvectors in order to determine the eigenvector with the largest correlation (in absolute value) with the first eigenvector from the original data. For large N (such as in the "Desmedt" example) it can be time consuming to compute the N eigenvectors for each bootstrap replication, however, we note the largest correlation in this data often

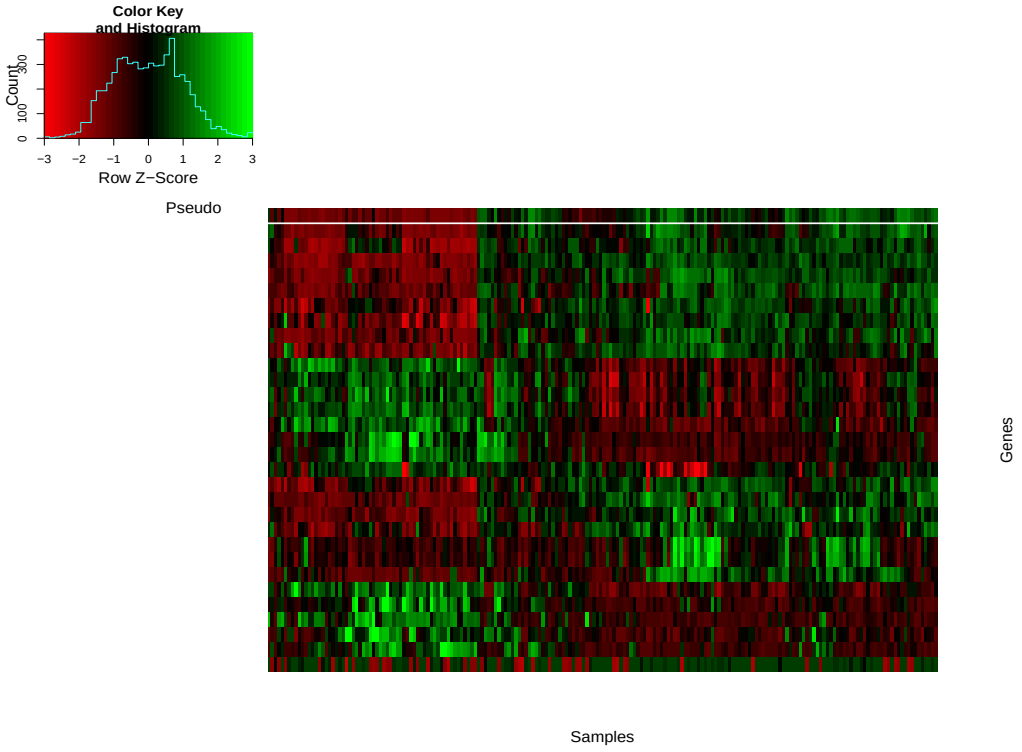

Figure 12: Breast Cancer Data: Heatmap of the top 30 genes (according to absolute eigenvalue) in the first functional pathway. The top row is the pseudo survival times while the other rows refer to the genes. For purposes of the color mapping, the data is row normalized and the gene rows and patient sample columns are reordered based on a dendogram using the Euclidean distance metric.

came from the first or second eigenvector. Hence, for speed in computation, for large N we propose only computing the first several eigenvectors rather than all N for Step 2(a).

Additionally we note the consistency diagnostic (5) is easily calculated and can be used to evaluate a candidate pathway determined by any discovery method. Although our code for analyzing the example data is available (see Additional Materials) for convenience, a formal `R` package implementing the method and future extensions is under development.

As noted in Section 2, W as shown in (7) can be thought of as two matrices multiplied together (element wise), the left matrix containing information on the modular structure of the genes and the right matrix containing information on the functional or relationship status of the genes with y. The multiplication of these matrices essentially combines the modularity property with the functional property. This is similar in spirit to one of the criteria proposed in Bair et al. (2006). In Section 6.6 of Bair et al. (2006), the authors propose examining the leading (largest) eigenvector of

$$Q(y, \alpha) = (1 - \alpha)XX^T + \alpha yy^T. \tag{10}$$

From studying (10) it also combines the modularity effect (XX^T) and the functional effect (yy^T). Note that Q in (10) is $n \times n$ which is a different dimension than W and that choosing α can be a challenge for a given dataset. It remains future work to compare our approach with the approach using (10).

Our simulations were for the high dimensional case where the sample size is less than the number of variables. The consistency of principal components of a covariance matrix has been established for high dimensionality when there are a few major eigenvalues (Hall, Marron, and Neeman, 2005, Ahn, Marron, Muller, and Chi, 2007). Future work will compare the statistical consistency of eigenvectors of W under similar conditions.

The methods described in this paper only determine which genes participate in a network. This is an easier task than determining the detailed graphical structure which entails as many parameters as there are pairs of variables. We have show that the less ambitious pathway variable identification methods can provide much information at relatively small sample sizes and low SNRs. Once the relevant genes are identified biological hypotheses may be suggested and more detailed network descriptions can be successful on the reduced set of variables (Edwards, 2000, Edwards, De Abreu, and Labouriau, 2010, Friedman, Hastie, and Tibshirani, 2008, Spirtes and Glymour, 1991).

Although we used correlation to represent the effect of a gene on y our method is easily adapted to more general measures of effect than x, y correlation. Likewise, other measures of association between genes with corresponding estimators can be substituted. The only requirement is that the measures be directional, i.e. distinguish positive and negative effects.

This paper has treated the case of a single functional pathway. Future work is directed toward extension to the case where multiple pathways are operating simultaneously, and identifying the null case where no functional module is present.

Additional materials

An R program for this study is available at

References

Ahn, J., J. Marron, K. M. Muller, and Y.-Y. Chi (2007): "The high-dimension, low-sample-size geometric representation holds under mild conditions," *Biometrika*, 94, 760–766.

Bair, E., T. Hastie, D. Paul, and R. Tibshirani (2006): "Prediction by supervised principal components," *Journal of the American Statistical Association*, 101.

Bair, E. and R. Tibshirani (2012): *superpc: Supervised principal components*, URL `http://CRAN.R-project.org/package=superpc`, r package version 1.09.

Breiman, L. (1996): "Bagging predictors," *Machine learning*, 24, 123–140.

Chou, T.-C. (1976): "Derivation and properties of michaelis-menten type and hill type equations for reference ligands," *Journal of theoretical biology*, 59, 253–276.

Csardi, G. and T. Nepusz (2006): "The igraph software package for complex network research," *InterJournal*, Complex Systems, 1695, URL `http://igraph.org`.

Desmedt, C., F. Piette, S. Loi, Y. Wang, F. Lallemand, B. Haibe-Kains, G. Viale, M. Delorenzi, Y. Zhang, M. S. d'Assignies, et al. (2007): "Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the transbig multicenter independent validation series," *Clinical cancer research*, 13, 3207–3214.

Edwards, D. (2000): *Introduction to graphical modelling*, Springer.

Edwards, D., G. C. De Abreu, and R. Labouriau (2010): "Selecting high-dimensional mixed graphical models using minimal aic or bic forests," *BMC bioinformatics*, 11, 18.

Friedman, J., T. Hastie, and R. Tibshirani (2008): "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, 9, 432–441.

Gautier, L., L. Cope, B. M. Bolstad, and R. A. Irizarry (2004): "affy—analysis of affymetrix genechip data at the probe level," *Bioinformatics*, 20, 307–315.

Gentleman, R. C., V. J. Carey, D. M. Bates, and others (2004): "Bioconductor: Open software development for computational biology and bioinformatics," *Genome Biology*, 5, R80, URL `http://genomebiology.com/2004/5/10/R80`.

Hall, P., J. Marron, and A. Neeman (2005): "Geometric representation of high dimension, low sample size data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 427–444.

Harary, F. et al. (1953): "On the notion of balance of a signed graph." *The Michigan Mathematical Journal*, 2, 143–146.

Hastie, T., R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani (2009): *The elements of statistical learning*, volume 2, Springer.

Horvath, S. (2011): *Weighted Network Analysis: Applications in Genomics and Systems Biology*, Springer Science & Business Media.

Jackson, D. A. (1995): "Bootstrapped principal components analysis- reply to mehlman et al." *Ecology*, 76, 644–645.

Juhasz, F. (1989): "On the theoretical backgrounds of cluster analysis based on the eigenvalue problem of the association matrix," *Statistics: A Journal of Theoretical and Applied Statistics*, 20, 573–581.

Klein, J. P., M. Gerster, P. K. Andersen, S. Tarima, and M. P. Perme (2008): "Sas and r functions to compute pseudo-values for censored data regression," *Computer methods and programs in biomedicine*, 89, 289–300.

Lee, T. I., N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, et al. (2002): "Transcriptional regulatory networks in saccharomyces cerevisiae," *Science*, 298, 799–804.

Maja Pohar Perme and M. Gerster (2012): *pseudo: Pseudo - observations*, URL `http://CRAN.R-project.org/package=pseudo`, r package version 1.1.

Miecznikowski, J. C., D. Wang, S. Liu, L. Sucheston, and D. Gold (2010): "Comparative survival analysis of breast cancer microarray studies identifies important prognostic genetic pathways," *BMC cancer*, 10, 573.

Milan, L. and J. Whittaker (1995): "Application of the parametric bootstrap to models that incorporate a singular value decomposition," *Applied Statistics*, 31–49.

Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon (2002): "Network motifs: simple building blocks of complex networks," *Science*, 298, 824–827.

Spirtes, P. and C. Glymour (1991): "An algorithm for fast recovery of sparse causal graphs," *Social Science Computer Review*, 9, 62–72.

Van den Bulcke, T., K. Van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal (2006): "Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms," *BMC bioinformatics*, 7, 43.

Wagner, G. P., M. Pavlicev, and J. M. Cheverud (2007): "The road to modularity," *Nature Reviews Genetics*, 8, 921–931.

Wright, S. (1934): "The method of path coefficients," *The Annals of Mathematical Statistics*, 5, 161–215.

Zhang, B. and S. Horvath (2005): "A general framework for weighted gene co-expression network analysis," *Statistical applications in genetics and molecular biology*, 4.