2016

A Weighted Gene Co-expression Network Analysis for Streptococcus sanguinis Microarray Experiments

Erik C. Dvergsten
Virginia Commonwealth University

A WEIGHTED GENE CO-EXPRESSION NETWORK ANALYSIS FOR *STREPTOCOCCUS SANGUINIS* MICROARRAY EXPERIMENTS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University

by

Erik Dvergsten
B.S. Mathematics and Biology, Christopher Newport University, 2013

Director: Nak-Kyeong Kim, Ph.D., Assistant Professor
Department of Biostatistics

Virginia Commonwealth University
Richmond, Virginia
July, 2016

Table of Contents

List of Figures iii

List of Tables iv

Abstract v

1. Biological Background **1**

1.1 Streptococcus sanguinis 1

1.2 Two-component Systems 1

2. Methods **2**

2.1 Weighted Gene Co-expression Network Analysis 2

2.2 Analysis 2

2.3 Gene Co-expression Similarity Measures and Adjacency Functions 3

2.4 Selecting Adjacency Function Parameters 4

2.5 Gene Module Identification 5

2.6 Scale-free Topology 5

2.7 Intramodular Connectivity and Module Membership 7

3. Results **8**

4. Discussion **31**

References **34**

Appendix: R Code **35**

List of Figures

Figure 1. Sample Clustering 8

Figure 2. A. Scale independence. **B**. Mean connectivity. 10

Figure 3. A. Histogram of connection frequency. **B.** Log-log plot of whole-network connectivity distribution..... 10

Figure 4. Gene Dendrogram.. 12

Figure 5. Network Heatmap Plot of All Genes..... 13

Figure 6. Eigengene Dendrogram 14

Figure 7. Eigengene Adjacency Heatmap.. 15

Figure 8. Absolute value of module membership vs. intramodular connectivity, separated by module 16

Figure 9. Absolute value of module membership raised to a power of 5 vs. intramodular connectivity, separated by module. 17

Figure 10. Network of the 30 most highly connected genes in the turquoise module 22

Figure 11. Network of the 30 most highly connected genes in the blue module..... 23

Figure 12. Network of the 30 most highly connected genes in the brown module. 24

Figure 13. Network of the 30 most highly connected genes in the yellow module 25

Figure 14. Network of the 30 most highly connected genes in the green module..... 26

Figure 15. Network of the 30 most highly connected genes in the red module 27

Figure 16. Network of the 30 most highly connected genes in the black module 28

Figure 17. Network of the 30 most highly connected genes in the pink module..... 29

Figure 18. Network of the 30 most highly connected genes in the magenta module..... 30

List of Tables

Table 1. Soft Threshold Fit Indices..... 11

Table 2. Summary of Module Assignments..... 12

Table 3. Top 30 most highly connected genes in the turquoise module. IMConnectivity is the TOM-based intramodular connectivity..... 22

Table 4. Top 30 most highly connected genes in the blue module. IMConnectivity is the TOM-based intramodular connectivity..... 23

Table 5. Top 30 most highly connected genes in the brown module. IMConnectivity is the TOM-based intramodular connectivity..... 24

Table 6. Top 30 most highly connected genes in the yellow module. IMConnectivity is the TOM-based intramodular connectivity..... 25

Table 7. Top 30 most highly connected genes in the green module. IMConnectivity is the TOM-based intramodular connectivity..... 26

Table 8. Top 30 most highly connected genes in the red module. IMConnectivity is the TOM-based intramodular connectivity..... 27

Table 9. Top 30 most highly connected genes in the black module. IMConnectivity is the TOM-based intramodular connectivity..... 28

Table 10. Top 30 most highly connected genes in the pink module. IMConnectivity is the TOM-based intramodular connectivity..... 29

Table 11. Top 30 most highly connected genes in the magenta module. IMConnectivity is the TOM-based intramodular connectivity..... 30

Abstract

A WEIGHTED GENE CO-EXPRESSION NETWORK ANALYSIS FOR *STREPTOCOCCUS SANGUINIS* MICROARRAY EXPERIMENTS

By Erik Dvergsten, B.S.

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

Virginia Commonwealth University, 2016

Director: Nak-Kyeong Kim, Ph.D., Assistant Professor, Department of Biostatistics

Streptococcus sanguinis is a gram-positive, non-motile bacterium native to human mouths. It is the primary cause of endocarditis and is also responsible for tooth decay. Two-component systems (TCSs) are commonly found in bacteria. In response to environmental signals, TCSs may regulate the expression of virulence factor genes.

Gene co-expression networks are exploratory tools used to analyze system-level gene functionality. A gene co-expression network consists of gene expression profiles represented as nodes and gene connections, which occur if two genes are significantly co-expressed. An adjacency function transforms the similarity matrix containing co-expression similarities into the adjacency matrix containing connection strengths. Gene modules were determined from the connection strengths, and various network connectivity measures were calculated.

S. sanguinis gene expression profile data was loaded for 2272 genes and 14 samples with 3 replicates each. The soft thresholding power $\beta = 6$ was chosen to maximize R^2 while maintaining a high mean number of connections. Nine modules were found. Possible meta-modules were found to be: Module 1: Blue & Green, Module 2: Pink, Module 3: Yellow, Brown &

Red, Module 4: Black, Module 5: Magenta & Turquoise. The absolute value of module membership was found to be highly positively correlated with intramodular connectivity. Each of the nine modules were examined. Two methods (intramodular connectivity and TOM-based connectivity followed by network mapping) for identifying candidate hub genes were performed. Most modules provided similar results between the two methods. Similar rankings between the two methods can be considered equivalent and both can be used to detect candidate hub genes. Gene ontology information was unavailable to help select a module of interest. This network analysis would help researchers create new research hypotheses and design experiments for validation of candidate hub genes in biologically important modules.

1. Biological Background

1.1. *Streptococcus sanguinis*

Streptococcus is a genus of spherical bacteria which are known to cause strep throat, scarlet fever, meningitis, pneumonia, and other infectious diseases [1]. *Streptococcus sanguinis* is a gram-positive, non-motile bacterium native to human mouths [2]. *S. sanguinis* can be found in the bloodstream leading to inhabitation of the heart valves. This can cause infective endocarditis, a potentially fatal heart disease. Entry into the bloodstream can occur during dental procedures as well as during routine eating [3]. *S. sanguinis* is recognized as the primary cause of infective endocarditis, and is also responsible for tooth decay via biofilm formation on tooth surfaces. *S. sanguinis* attaches directly to oral surfaces which allows for the subsequent attachment of various other microorganisms which contribute to dental plaque formation, tooth decay and periodontal disease [2]. The SK36 strain of *S. sanguinis* has 2270 purported protein coding genes, and contains 2.39 million bases [1].

1.2. Two-component systems

Two-component systems (TCSs) are commonly found in bacteria [4]. Genes involved in signal transduction via TCSs have been found to be virulence factors in disease models [4]. TCSs are comprised of histidine kinase, a membrane-bound protein, and a cytosolic response regulator protein [4]. TCSs are known to modify photosynthesis, pathogenicity, osmoregulation, and other responses [4]. In response to environmental signals, TCSs may regulate the expression of virulence factor genes [4].

2. Methods

2.1. Weighted Gene Co-expression Networks

Gene co-expression networks are exploratory tools used to analyze system-level gene functionality. A gene co-expression network consists of gene expression profiles represented as nodes and gene connections, which occur if two genes are significantly co-expressed (determined by pairwise gene expression correlations) [5]. Modules are clusters of highly interconnected genes (i.e. highly correlated genes). A module eigengene is considered representative of the gene expression profiles in a given module, and by definition, is the first principal component of that module [5]. The first principal component explains the highest proportion of the variance among genes. When constructing a network, a hard threshold is implemented for defining unweighted network connections, while for weighted networks, a soft threshold is applied to assign each pair of genes a connection weight [5].

2.2. Analysis

Statistical analyses were performed using the R package WGCNA, a weighted correlation network analysis tool developed specifically for analyzing large, high-dimensional genetic datasets [6]. Network construction, module detection, topological overlap matrix construction, and various plots were produced with WGCNA (Figure 1 – Figure 9). VisANT, a biological network visualization tool, was used to produce the network visualizations in Figure 10 – Figure 18 [7].

2.3. Gene Co-expression Similarity Measures and Adjacency Functions

Every co-expression network relates to an adjacency matrix. The adjacency matrix is used to define node connectivity and houses connection strengths between node pairs [5]. The $n \times n$ adjacency matrix $A = [a_{ij}]$ is constructed from an $n \times n$ similarity matrix $S = [s_{ij}]$, which measures the level of similarity between gene expression profiles across experiments [5]. Define the similarity matrix *S* as the absolute value of the Pearson correlation between each pair of genes i and j : $s_{ij} = |cor(i,j)|, for\ s_{ij} \in [0,1]$. The diagonal elements a_{ii} are conventionally defined as 0, and $a_{ij} \in [0,1]$ for weighted networks [5]. Contrastingly, the adjacency matrix of unweighted networks is a binary system of a value 1 for being connected, and 0 for being unconnected [5].

An adjacency function transforms the similarity matrix containing co-expression similarities into the adjacency matrix containing connection strengths [5]. The choice of adjacency function is determined by the weight properties of the network. The term weight properties references whether a network is weighted or unweighted. Unweighted networks apply hard thresholding using the signum function

$$a_{ij} = signum(s_{ij}, \tau) = \begin{cases} 1\ if\ s_{ij} \geq \tau \\ 0\ if\ s_{ij} < \tau \end{cases}$$

which presents intuitive networks (i.e. the number of direct neighbors equals the node connectivity) [5]. However, this can present a problem. For example, if the threshold τ is 0.75 and the similarity is 0.74, the connection does not occur and consequently information is lost.

Additionally, node connectivity using hard thresholding is sensitive to the choice of the threshold [8].

Soft thresholding helps avoid these disadvantages by defining a power adjacency function:

$$a_{ij} = \left|s_{ij}\right|^{\beta}, for\ \beta > 1.$$

β is selected to approximate scale-free topology, which will be introduced in section 2.6 [5].

2.4. Selecting Adjacency Function Parameters

There are a few factors to consider when determining the parameters of the adjacency function since the parameters determine the sensitivity and specificity of the pairwise connection strengths [5]. Network connectivity k_i of node i is defined as the number of its direct connections with other nodes. A similar Topological Overlap Matrix (TOM) based connectivity ω_i can also be used [5]. Let

$$\omega_{ij} = \frac{\sum_u a_{iu}a_{uj} + a_{ij}}{\min\{k_i, k_j\} + 1 - a_{ij}}$$

where $k_i = \sum_u a_{iu}$ and $k_j = \sum_u a_{ju}$. Then

$$\omega_i = \sum_{j=1}^{n} \omega_{ij}$$

It has been shown that a TOM-based measure of connectivity ω_i is superior to the standard k_i measure [5]. The topological overlap matrix $\Omega = [\omega_{ij}]$ is transformed into a

dissimilarity matrix defined by $d_{ij} = 1 - \omega_{ij}$, which is subsequently used for clustering gene expression profiles [5].

2.5. Gene Module Identification

The next step after network construction is module detection. Modules are clusters of closely interconnected nodes (i.e. genes with high topological overlap). A weighted topological overlap measure has been shown to deliver more interconnected modules than an unweighted measure [5]. The WGCNA package uses unsupervised clustering to identify gene modules. Using a TOM-based dissimilarity, average linkage hierarchical clustering is performed using the standard R function *hclust*. Modules are depicted as dendrogram branches, and cutting is performed using the dynamic hybrid tree cut algorithm [6].

A TOM plot is a color-coded matrix representation of a summary of the co-expression network, which depicts the values of the dissimilarity matrix. Rows and columns are sorted by the hierarchical clustering dendrogram. Red and yellow indicate low and high dissimilarity respectively (see Figure 5). Modules are described as red squares along the diagonal. Note that TOM plots are symmetric along the diagonal because they are graphical representations of the topological overlap matrix which is also symmetric.

2.6. Scale-free Topology

The evolution of biological systems is thought to be driven by a power-law distribution [9] [10]. In a power-law distribution, scale-free topology is directly related to the growth of the

network [5]. New nodes prefer to connect with existing nodes. An essential property of a scale-free network is that the frequency distribution $p(k)$ of the connectivity follows a power law:

$$p(k) \sim k^{-\gamma}.$$

The Pearson correlation measures the strength of the linear relationship between two variables. R^2, the square of the Pearson correlation of the aforementioned regression, can be used to show the degree to which a network satisfies scale-free topology [5]. An R^2 approaching 1 will approximate a straight line which signifies a good fit for the data, and indicates a scale-free topology [5]. A simple linear regression of the log-log relationship yields a straight line and a perfect fit of the data when R^2 is 1. Thus, a straight line on a plot of $log_{10}\, p(k)$ versus $log_{10}\, k$ demonstrates scale-free topology (see Figure 3). This scale-free topology relationship can be generalized as

$$\log_{10} p(k) = \beta_0 + \beta_1 log_{10} k.$$

Trade-offs exist between maximizing R^2 and retaining a high mean number of connections. Only adjacency function parameter values leading to a scale-free topology fitting index of $R^2 > 0.80$ should be considered [5]. The mean connectivity should also be high enough to contain enough information for module detection [5]. A gene co-expression network that does not approximately satisfy scale-free topology is considered biologically suspicious and therefore should not be used [5].

The topology of scale-free networks is largely controlled by highly connected nodes called hubs [5]. One of the implications of scale-free topology is that relatively few hubs exist in the network. Less-connected nodes are linked to the hub, forming a network. An important

characteristic of scale-free networks is that they have a relatively large error tolerance [5]. Simple single celled organisms such as *S. sanguinis* reproduce and thrive despite undergoing severe environmental or pharmaceutical interventions, which is thought to be credited to the high error tolerance of a metabolic scale-free network [5].

2.7. Intramodular Connectivity and Module Membership

Network connectivity can also be defined with respect to each individual module. The intramodular connectivity $k.in$ and the TOM-based intramodular connectivity $\omega.in$ and can be defined similarly to whole-network connectivity [5]. Intramodular connectivity measures have been shown to be more biologically significant than whole-network connectivity [5].

Hierarchical clustering results in a binary module assignment [6]. Each gene is either in a particular module, or not in that module. Therefore, it can be beneficial to define a continuous measure of uncertainty of module assignment. This is valuable for genes that are near module boundaries [6]. The module membership of gene i in module q can be defined as

$$K_{cor,i}^{(q)} = cor(x_i, E^{(q)}),$$

where x_i is the expression profile of gene i and $E^{(q)}$ is the module eigengene of module q [6]. Since module membership is a measure of correlation, $K_{cor,i}^{(q)}$ is in [-1,1], and as the absolute value of $K_{cor,i}^{(q)}$ increases, the similarity between gene i and the module eigengene of module q increases [6]. The relationship between module membership and intramodular connectivity can be seen in Figure 8 and Figure 9.

3. Results

S. sanguinis gene expression profile data was loaded for 2272 genes and 14 samples with 3 replicates each. Data has previously been normalized. Genes and samples were checked for excessive missing values. 538 genes with excessive missing values were removed, leaving 1734 genes. Next, a dendrogram was created via average linkage hierarchical clustering to detect sample outliers (see Figure 1). After standardizing the connectivity, one sample was found to be an outlier. However, the standardized connectivity of this sample was only borderline outlying, so the sample was included in all subsequent analyses.

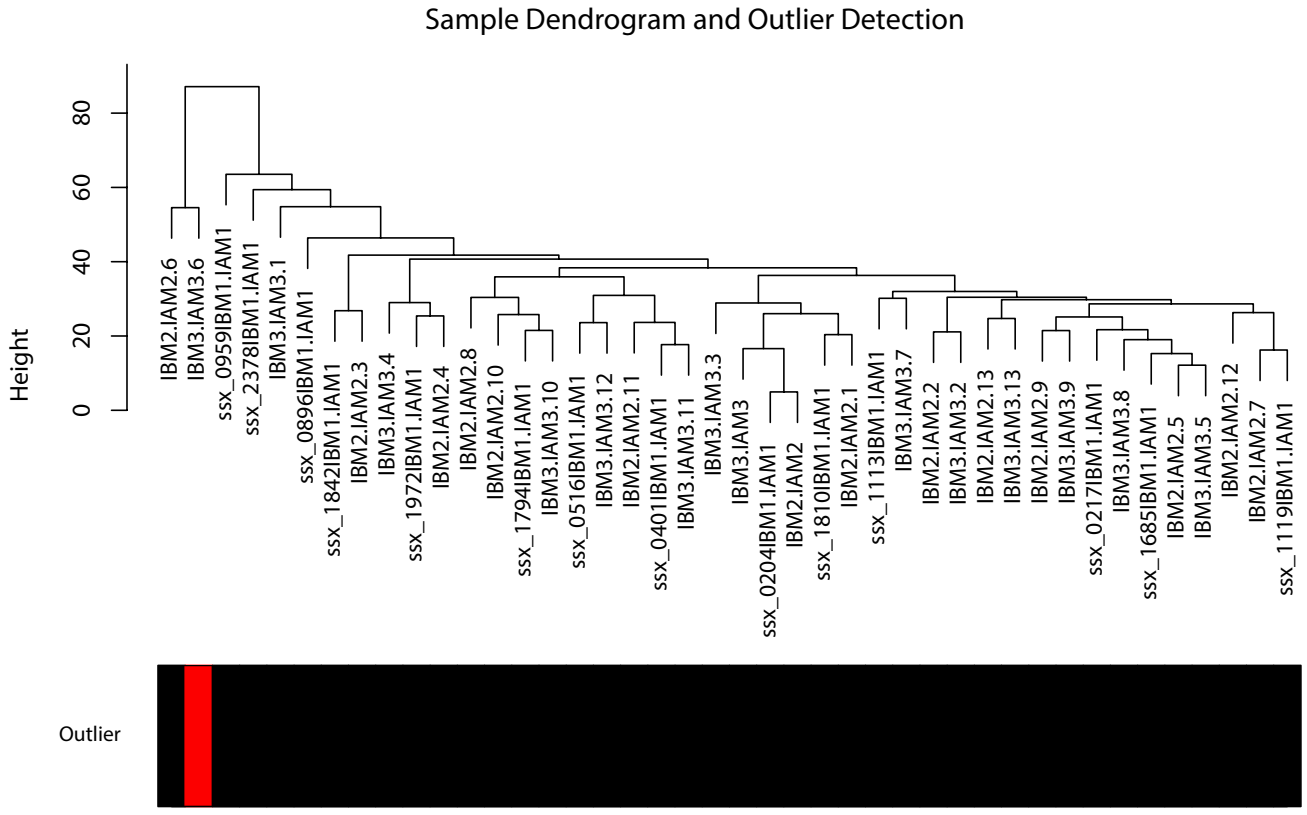

Figure 1. Sample Clustering.

Using the scale-free topology criteria from Section 2.6, a soft thresholding power $\beta = 6$ was chosen to best approximate scale-free topology. This choice gives $R^2 = 0.966$ and the mean number of connections $k = 18.01$ (see Table 1). Figure 2A plots scale-free topology model fit R^2 versus the candidate soft threshold powers. Figure 2B plots the mean connectivity versus the candidate soft threshold powers. Figure 3 contains a histogram of the frequency of connections and a plot assessing scale-free topology. A highly skewed histogram is said to approximate a scale-free network [5].

The soft thresholding power $\beta = 6$ was chosen to maximize R^2 while maintaining a high mean number of connections. As shown in Figure 2A, $\beta = 6$ does not maximize R^2 or maximize the mean number of connections. The mean number of connections is a strictly monotonic function, but R^2 is not. However, Figure 2A is monotonically increasing until $\beta = 7$. At approximately $R^2 = 0.96$, Figure 2A levels out. The scale-free topology fit index ($R^2 = 0.967$) at $\beta = 7$ is greater than at $\beta = 6$ ($R^2 = 0.966$), but this small increase in R^2 results a comparatively large drop in the mean number of connections ($k = 18.01$ for $\beta = 6, k = 12.87$ for $\beta = 7$). Similarly, a slightly larger increase in scale-free topology fit index ($R^2 = 0.976$ for $\beta = 9$), reduces the mean number of connections too much ($k = 7.60$). For these reasons, $\beta = 6$ was chosen. The table of soft threshold fit indices can be found in Table 1.

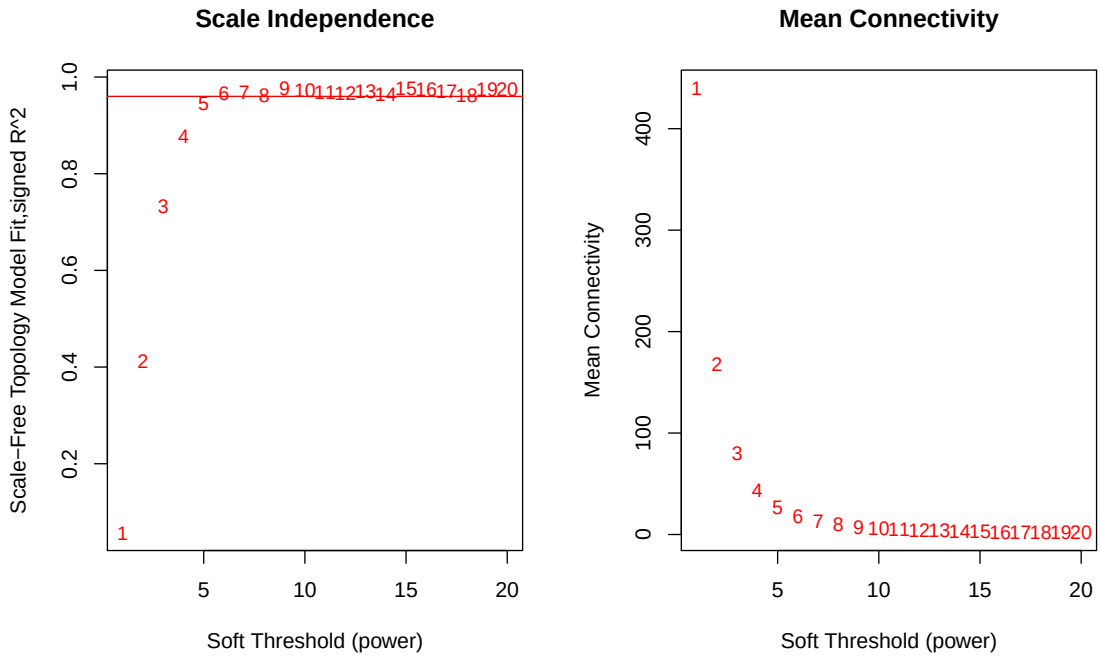

Figure 2. **A.** Scale independence. **B**. Mean connectivity.

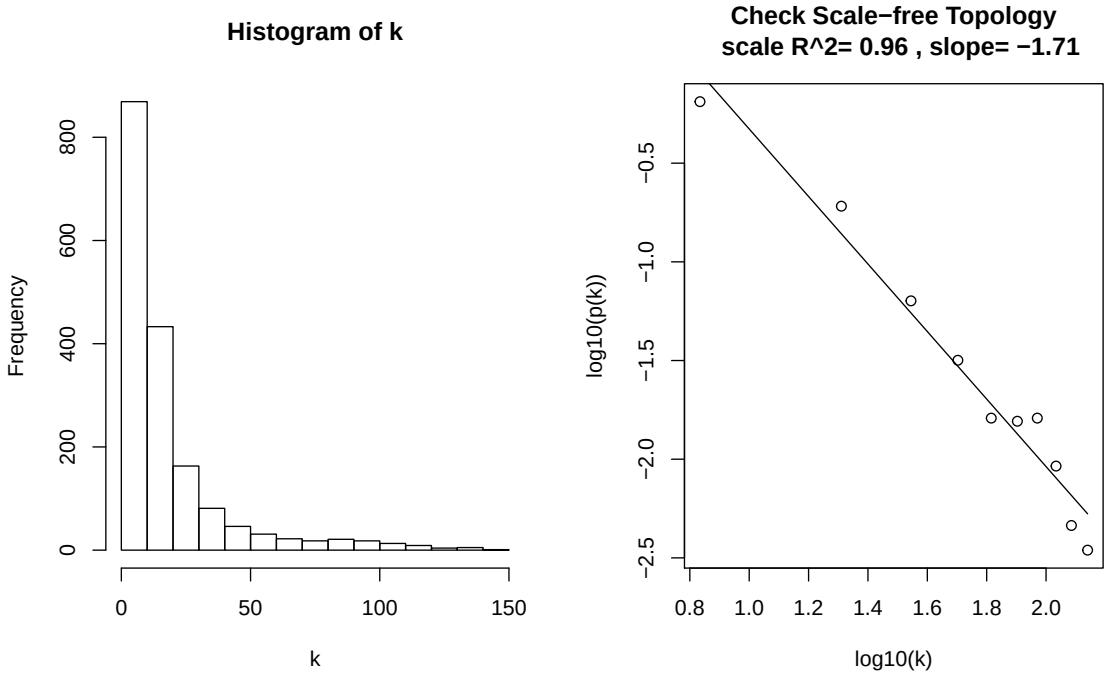

Figure 3. **A.** Histogram of connection frequency. **B.** Log-log plot of whole-network connectivity distribution.

Power	SFT.R.sq	slope	truncated.R.sq	mean.k.	median.k.	max.k.
1	0.06	-0.87	0.97	440.20	432.62	736.13
2	0.41	-1.56	0.98	168.43	156.61	424.07
3	0.73	-1.78	0.99	79.96	67.91	287.36
4	0.88	-1.87	0.98	43.95	32.86	215.63
5	0.95	-1.80	0.98	26.96	17.56	172.98
6	0.97	-1.71	0.98	18.01	9.96	145.08
7	0.97	-1.61	0.96	12.87	6.05	125.41
8	0.96	-1.55	0.95	9.68	3.78	110.73
9	0.98	-1.45	0.97	7.60	2.48	99.25
10	0.97	-1.39	0.97	6.16	1.66	89.96
11	0.97	-1.36	0.96	5.12	1.13	82.89
12	0.97	-1.31	0.96	4.34	0.79	76.93
13	0.97	-1.28	0.97	3.74	0.56	71.75
14	0.96	-1.26	0.96	3.27	0.41	67.22
15	0.98	-1.24	0.97	2.89	0.30	63.19
16	0.97	-1.23	0.97	2.58	0.22	59.59
17	0.97	-1.21	0.97	2.32	0.17	56.35
18	0.96	-1.20	0.96	2.10	0.13	53.41
19	0.98	-1.19	0.97	1.92	0.10	50.74
20	0.98	-1.18	0.97	1.75	0.08	48.29

Table 1. Soft Threshold Fit Indices.

Figure 4 displays the dendrogram created from gene clustering and the corresponding color module memberships. Modules were identified using a minimum module size of 30, and similar modules were merged if the cut height was less than 0.25. The cut height is determined using the dynamic hybrid tree cut algorithm [11]. Table 2 reports nine modules such as the turquoise module with 519 genes and the magenta module with 46 genes. 140 genes were outside of those nine modules and are labeled as the grey module.

Module Color	Total Genes Assigned
Turquoise	517
Blue	243
Brown	242
Yellow	183
Green	138
Red	80
Black	74
Pink	71
Magenta	46
Grey	140

Table 2. Summary of Module Assignments. Grey signifies genes that were not assigned to any module.

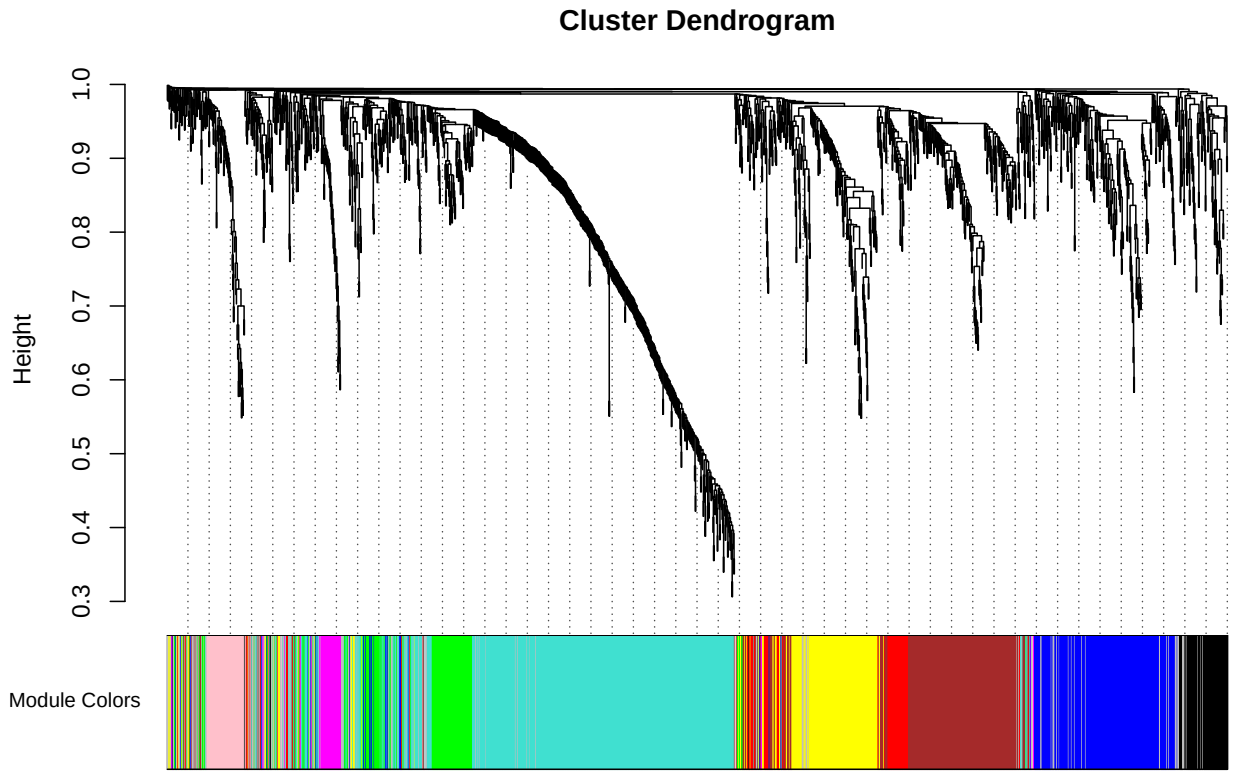

Figure 4. Gene Dendrogram. Colors along the bottom indicate module assignment. Grey bars signify genes that were not assigned to any module.

A network topological overlap heatmap was produced from TOM-based dissimilarity measures. The resulting TOM plot with dendrogram and module membership colors is found in Figure 5. It is apparent that the lower the genes merge in the clustering, the darker the color on the heatmap and consequently the higher the topological overlap. Since modules are defined by a branch cut height, a gene that is found towards the tip of a branch is more likely to belong in its assigned module compared to a gene that is higher up in the tree. These genes are said to have higher module membership.

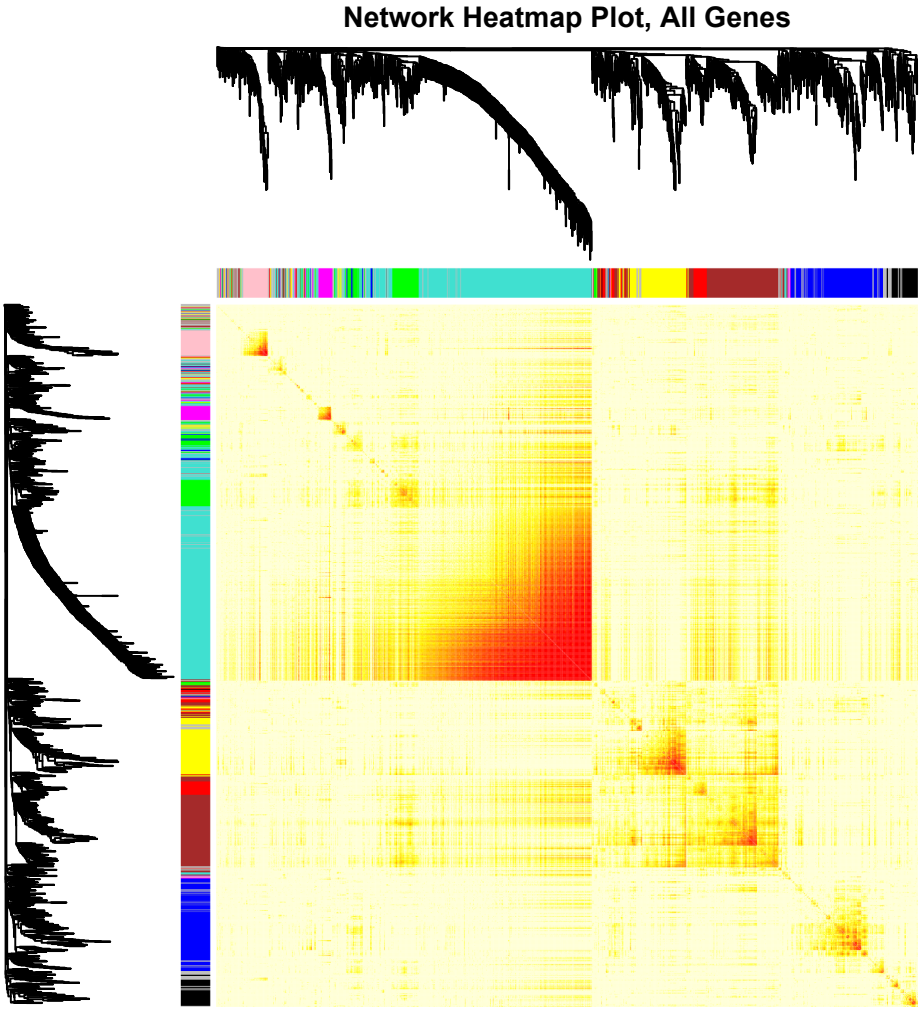

Figure 5. Network Heatmap Plot of All Genes. Each row and column corresponds to a gene. Light colors signify low topological overlap, while dark red represents high topological overlap. Dark squares along the diagonal represent modules. The gene dendrogram with module assignment are included along the axes.

The eigengene dendrogram (Figure 6) and the eigengene adjacency heatmap (Figure 7) depict tight clusters of correlated eigengenes called “meta-modules” [6]. The brown, red, and yellow modules are highly positively related. The blue and green modules are moderately to highly positively related. The magenta and turquoise modules are clustered together, but only moderately related. Meta-modules can also be seen as groups of reddish squares along the diagonal (Figure 7). The meta-modules (Module 1: Blue & Green, Module 2: Pink, Module 3: Yellow, Brown & Red, Module 4: Black, Module 5: Magenta & Turquoise) can clearly be seen in Figure 6 and Figure 7.

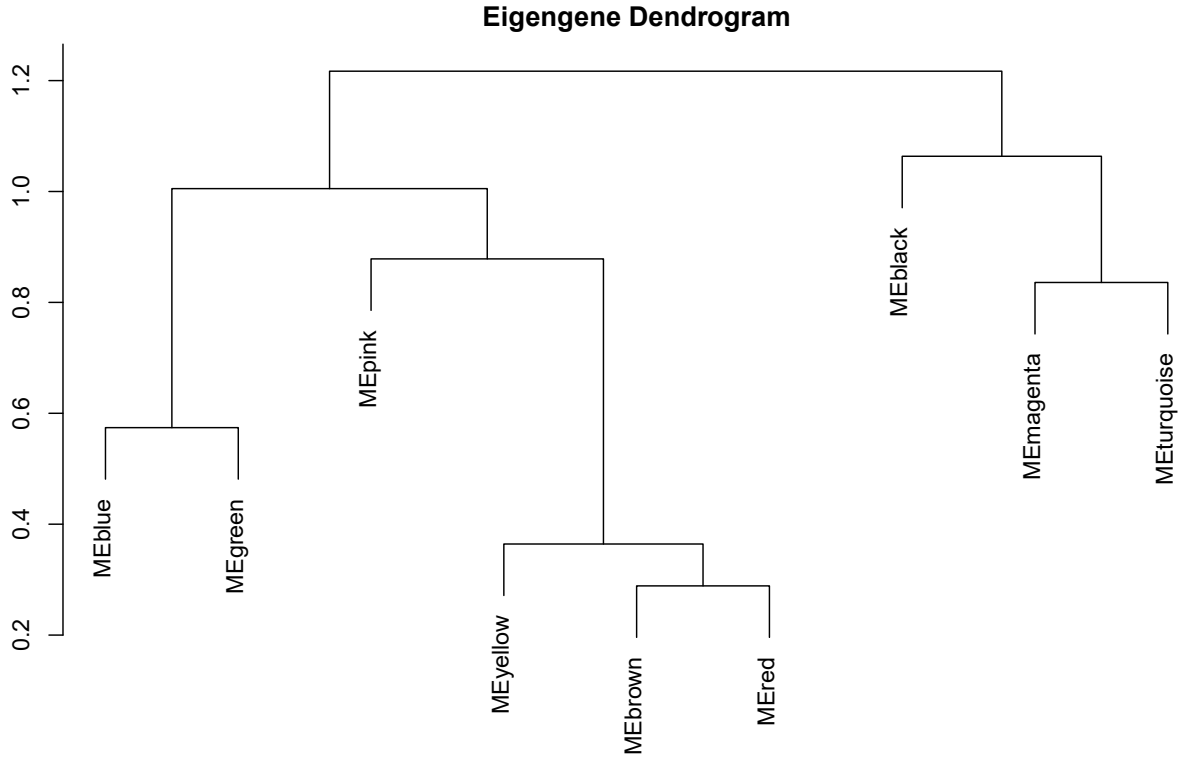

Figure 6. Eigengene Dendrogram. The brown, red, and yellow modules are highly related. The blue and green modules are also related. The magenta and turquoise modules are also related.

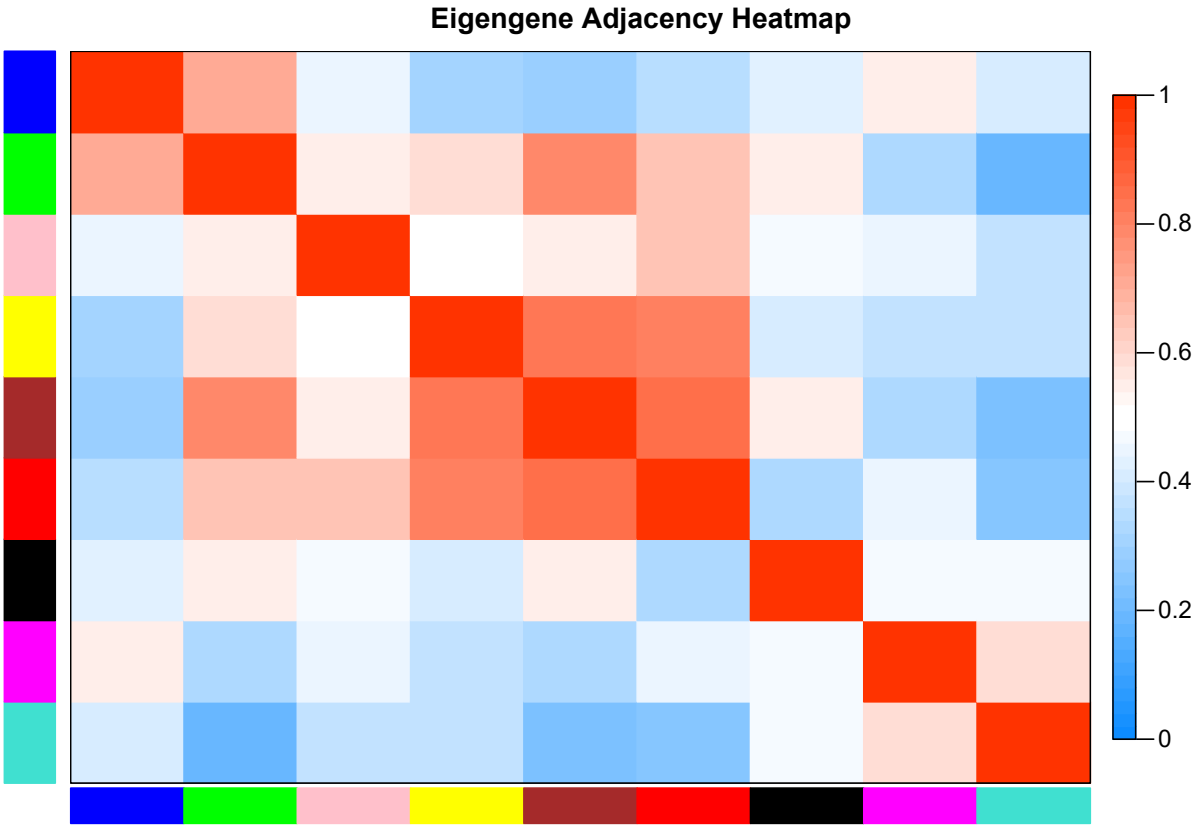

Figure 7. Eigengene Adjacency Heatmap. Each row and column correspond to one module eigengene labeled with its corresponding color. Red represents positive correlation and high adjacency, and blue represents negative correlation and low adjacency. White represents zero correlation and an adjacency of 0.5.

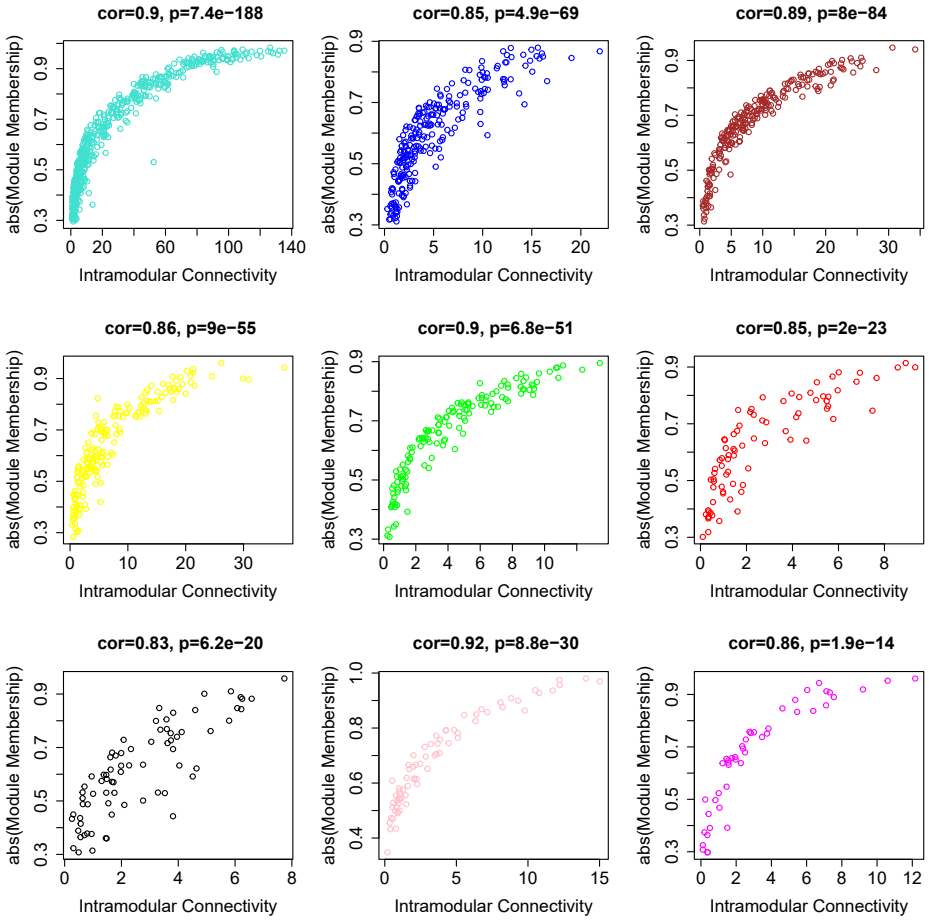

Figure 8. Absolute value of module membership vs. intramodular connectivity, separated by module.

The absolute value of module membership is highly positively correlated with intramodular connectivity (see Figure 8). When the absolute value of module membership is transformed by a power of 5, there is a high positive linear correlation between intramodular connectivity and the module membership in all nine modules (see Figure 9). In general, a high module membership corresponds to a high intramodular connectivity. Genes with both of these properties may be important candidate hub genes. Since the absolute value of modular membership and intramodular connectivity are highly correlated, either measure may be useful in selecting candidate hub genes.

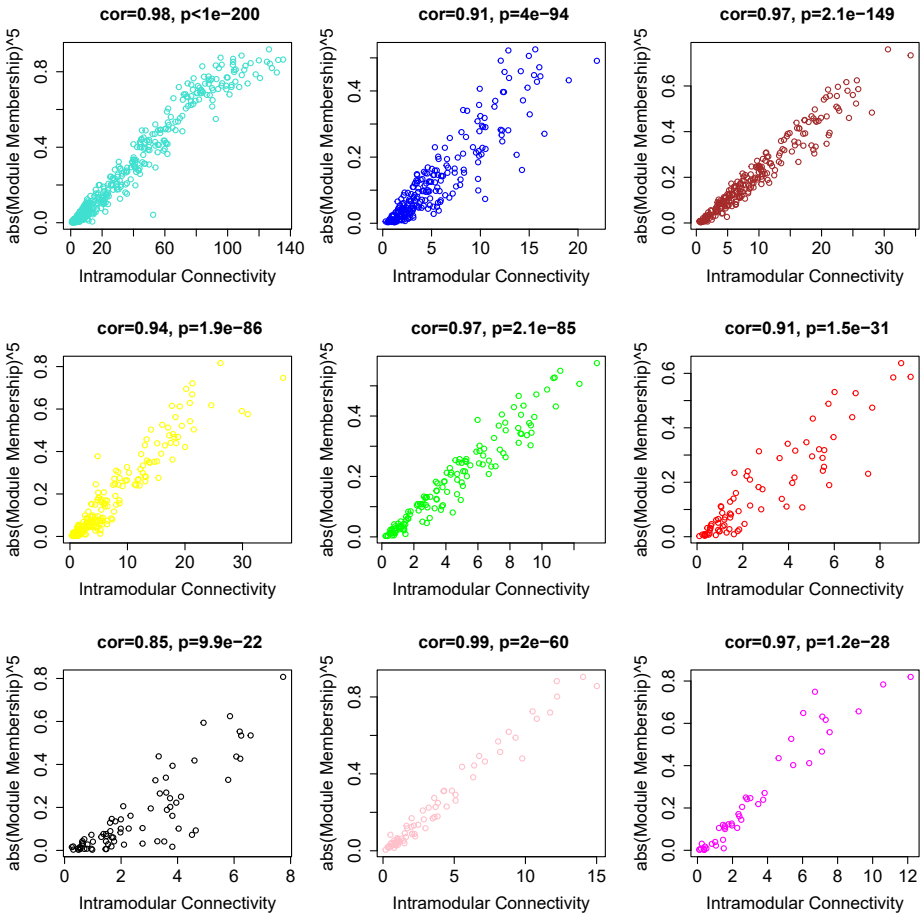

Figure 9. Absolute value of module membership raised to a power of 5 vs. intramodular connectivity, separated by module.

Important modules may want to be examined in a web chart to visually see the most highly connected genes and their corresponding connections. Therefore, network information can be exported from R and visualized in VisANT, a biological network visualization tool [7]. The 30 most highly connected genes based on intramodular connectivity are presented for each of the nine modules (see Table 3 – Table 11). Additionally, a network of the 30 most highly connected genes are displayed for each of the nine modules. Only the connections between genes with the highest topological overlap are displayed (see Figure 10 – Figure 18). This ad hoc threshold was determined visually to increase readability as well as present a similar number of top hub genes per module (ranges from 4 to 6) and a similar number of minimum connections to

be considered a top hub. The candidate hub genes (Figure 10 – Figure 18) are not always the most highly connected genes based on TOM-based intramodular connectivity (Table 3 – Table 11), but some overlap exists. For simplicity in the following analyses, hub genes refer to the most highly connected genes from Figure 10 – Figure 18.

The turquoise module has 517 total genes. The top 30 genes are listed in Table 3 and the network is displayed in Figure 10. The intramodular connectivity ranges from 99.19 to 135.36 in the top 30 genes. After filtering the number of connections with a topological overlap threshold of 0.56, the five genes with at least 12 connections each were defined as hub genes. Interestingly, these five hub genes are not the exact same top five hub genes calculated from TOM-based intramodular connectivity. For example, the gene with the highest intramodular connectivity (SSA_1599) was not defined as a hub gene under these conditions. However, SSA_2295 was defined as a hub gene, but is only the 28th most interconnected. Nineteen of the top 30 genes in the turquoise module are hypothetical proteins. The hub gene with the highest intramodular connectivity is SSA_1336, an Ankyrin repeat-containing protein gene.

The blue module has 243 total genes. The top 30 genes are listed in Table 4 and the network is displayed in Figure 11. The intramodular connectivity ranges from 10.52 to 21.93 in the top 30 genes. After filtering the number of connections with a topological overlap threshold of 0.18, the four genes with at least 11 connections each were defined as hub genes. As in the turquoise module, these four hub genes are not the exact same top four hub genes calculated from TOM-based intramodular connectivity. The only difference between these sets is the hub gene SSA_1662. This is the 24th highest interconnected gene. SSA_0424, a putative exopolysaccharide biosynthesis protein, is the 3rd most interconnected gene, but is not defined

as a hub gene under these conditions. The hub gene with the highest intramodular connectivity is SSA_0425, a glycosyltransferase gene.

The brown module contains 242 genes. The top 30 genes are listed in Table 5 and the network is presented in Figure 12. The intramodular connectivity ranges from 18.84 to 34.22 in the top 30 genes. After filtering the number of connections with a topological overlap threshold of 0.21, the five genes with at least 14 connections each were defined as hub genes. As with the previous modules, the five hub genes are not the exact same top five hub genes calculated from TOM-based intramodular connectivity. Interestingly, the two genes (SSA_1260 and SSA_1261) with the highest intramodular connectivity are sequential. These two genes are also hub genes. There are two other sequential hub genes, SSA_0524 and SSA_0525; both are putative microcompartment protein genes. They are 6 and 8 respectively in Table 5.

The yellow module contains 183 total genes. The top 30 genes are listed in Table 6 and the network is displayed in Figure 13. The intramodular connectivity ranges from 15.15 to 37.06 in the top 30 genes. After filtering the number of connections with a topological overlap threshold of 0.30, the five genes with at least 10 connections each were defined as hub genes. Three sequential hub genes (SSA_0737, SSA_0738, and SSA_0739) exist in the yellow module and are 24, 14, and 3 respectively in Table 6. The two genes with the highest intramodular connectivity are also hub genes (SSA_1591, a putative dipeptidase, and SSA_1695, a BglG family transcriptional antiterminator).

The green module has 138 genes. The top 30 genes are listed in Table 7 and the network is displayed in Figure 14. The intramodular connectivity ranges from 7.10 to 13.43 in the top 30 genes. After filtering the number of connections with a topological overlap threshold of 0.11, the

five genes with at least 14 connections each were defined as hub genes. 12 of the top 30 genes are ribosomal subunit protein genes (eight 50S large subunits, four 30S small subunits). Three hub genes are 50S ribosomal protein genes (SSA_0108, SSA_0110, and SSA_0112), and are all in the top 10 interconnected genes in the green module. SSA_0116, a 30S ribosomal protein gene is also a hub gene. The gene with the highest intramodular connectivity (SSA_1508, putative ABC-type lipopolysaccharide transport system, and permease component) is also a hub gene.

The red module has 80 genes. The top 30 genes are listed in Table 8 and the network is displayed in Figure 15. The intramodular connectivity ranges from 2.71 to 9.34 in the top 30 genes. After filtering the number of connections with a topological overlap threshold of 0.11, the four genes with at least 10 connections each were defined as hub genes. Three hub genes (SSA_0473, SSA_0488, and SSA_0491) are three of the top four most highly interconnected genes. The 4th hub gene (SSA_0483, a putative siroheme synthase gene) is the 6th most highly interconnected gene in the red module.

The black module has 74 genes. The top 30 genes are listed in Table 9 and the network is displayed in Figure 16. The intramodular connectivity ranges from 3.05 to 7.74 in the top 30 genes. After filtering the number of connections with a topological overlap threshold of 0.11, the five genes with at least 12 connections each were defined as hub genes. Two sequentially named genes are hub genes (SSA_1631 and SSA_1632). These hubs are also in the top 5 interconnected genes in the black module. All five hubs are in the top 7 interconnected genes. 13 of the top 30 genes are hypothetical protein genes.

The pink module has 71 genes. The top 30 genes are listed in Table 10 and the network is displayed in Figure 17. The intramodular connectivity ranges from 3.20 to 15.02 in the top 30

genes. After filtering the number of connections with a topological overlap threshold of 0.30, the six genes with at least 12 connections each were defined as hub genes. The top 6 genes with the highest intramodular connectivity are also the 6 hub genes. The pink module is the only module where this is the case. Four hub genes are sequentially named (SSA_0675, SSA_0676, SSA_0677, and SSA_0678).

The magenta module has 46 genes. The top 30 genes are listed in Table 11 and the network is displayed in Figure 18. The intramodular connectivity ranges from 1.55 to 12.17 in the top 30 genes. After filtering the number of connections with a topological overlap threshold of 0.19, the five genes with at least 12 connections each were defined as hub genes. Five of the top six most highly interconnected genes are hub genes. Ten hypothetical protein genes are scattered throughout the top 30 interconnected genes.

	IMConnectivity	Genes
1	135.36	SSA_1599->hypothetical_protein
2	132.24	SSA_1336->ankyrin_repeat-containing_protein
3	131.34	SSA_1473->hypothetical_protein
4	128.19	SSA_0557->hypothetical_protein
5	126.91	SSA_0949->hypothetical_protein
6	126.33	SSA_1627->hypothetical_protein
7	125.04	SSA_0561->RNA:NAD_2'-phosphotransferase,_putative
8	120.63	SSA_2389->arsenical_resistance_operon_transcription_repressor,_putative
9	119.89	SSA_1332->hypothetical_protein
10	117.89	SSA_1331->hypothetical_protein
11	114.64	SSA_2067->hypothetical_protein
12	112.73	SSA_0560->hypothetical_protein
13	111.38	SSA_2388->hypothetical_protein
14	110.61	SSA_1474->putative_lipoprotein
15	109.02	SSA_0296->XRE_family_transcriptional_regulator
16	108.87	SSA_1489->hypothetical_protein
17	107.39	SSA_0559->hypothetical_protein
18	105.16	SSA_1284->hypothetical_protein
19	104.97	SSA_2384->acetyltransferase
20	104.22	SSA_2251->hypothetical_protein
21	104.15	SSA_0750->hypothetical_protein
22	103.52	SSA_2383->prophage_maintenance_system_killer_protein_(DOC:_death-on-curing),_putative
23	103.48	SSA_0880->hypothetical_protein
24	103.34	SSA_1337->hypothetical_protein
25	102.48	SSA_0699->methyltransferase,_putative
26	102.45	SSA_1334->hypothetical_protein
27	101.21	SSA_1315->hypothetical_protein
28	99.97	SSA_2295->phage_integrase_family_integrase/recombinase
29	99.45	SSA_2187->membrane_associated_protein
30	99.19	SSA_0558->cytosolic_protein,_putative

Table 3. Top 30 most highly connected genes in the turquoise module. IMConnectivity is the TOM-based intramodular connectivity.

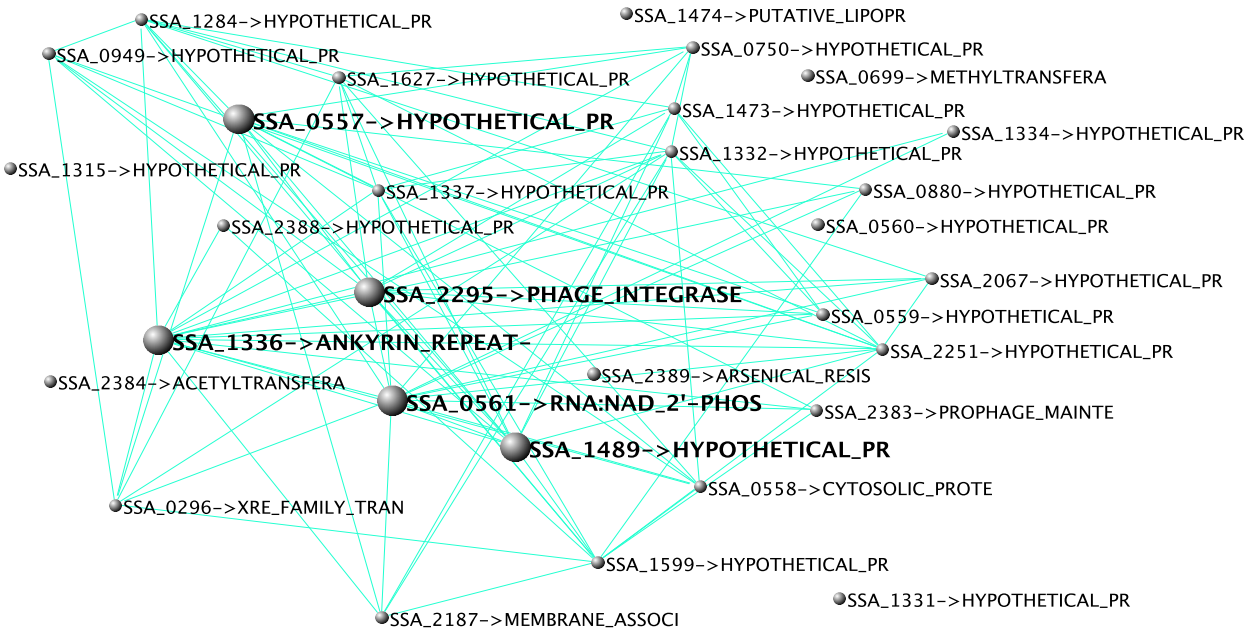

Figure 10. Network of the 30 most highly connected genes in the turquoise module. Connections displayed correspond to a topological overlap greater than 0.56. The top 5 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 12 connections in this subset.

	IMConnectivity	Genes
1	21.93	SSA_0425->glycosyltransferase
2	19.05	SSA_0728->protease,_putative
3	16.56	SSA_0424->exopolysaccharide_biosynthesis_protein,_putative
4	16.14	SSA_0958->hypothetical_protein
5	16.04	SSA_2006->4-methyl-5(B-hydroxyethyl)-thiazole_monophosphate_biosynthesis_enzyme,_putative
6	15.80	SSA_1982->LytR/AlgR_family_transcriptional_regulator_putative
7	15.62	SSA_2130->hypothetical_protein
8	15.50	SSA_1981->hypothetical_protein
9	15.06	SSA_0704->isocitrate_dehydrogenase_[NADP],_putative
10	14.97	SSA_0178->UDP-N-acetylglucosamine_2-epimerase,_putative
11	14.83	SSA_0702->aconitate_hydratase
12	14.36	SSA_0703->citrate_synthase
13	14.26	SSA_1481->FmtA-like_protein,_putative
14	14.14	SSA_1201->phosphopantothenate–cysteine_ligase
15	13.71	SSA_1480->hypothetical_protein
16	12.94	SSA_2131->DNA-binding_protein,_putative
17	12.87	SSA_0181->glycosyl_transferase_family_protein
18	12.86	SSA_0729->hypothetical_protein
19	12.47	SSA_0180->hypothetical_protein
20	12.42	SSA_0183->hypothetical_protein
21	12.31	SSA_0182->endoglucanase,_putative
22	12.24	SSA_2009->heat-inducible_transcription_repressor
23	12.23	SSA_2349->dTDP-4-dehydrorhamnose_3,5-epimerase,_putative
24	12.18	SSA_1662->NADH-dependent_oxidoreductase,_putative
25	12.09	SSA_0460->multiple_antibiotic_resistance_operon_transcription_repressor_(MarR),_putative
26	12.08	SSA_0959->two-component_response_transcriptional_regulator
27	11.53	SSA_1015->prenyltransferase
28	11.44	SSA_1202->phosphopantothenoylcysteine_decarboxylase
29	10.53	SSA_2342->SPX_domain-containing_protein
30	10.52	SSA_1773->hypothetical_protein

Table 4. Top 30 most highly connected genes in the blue module. IMConnectivity is the TOM-based intramodular connectivity.

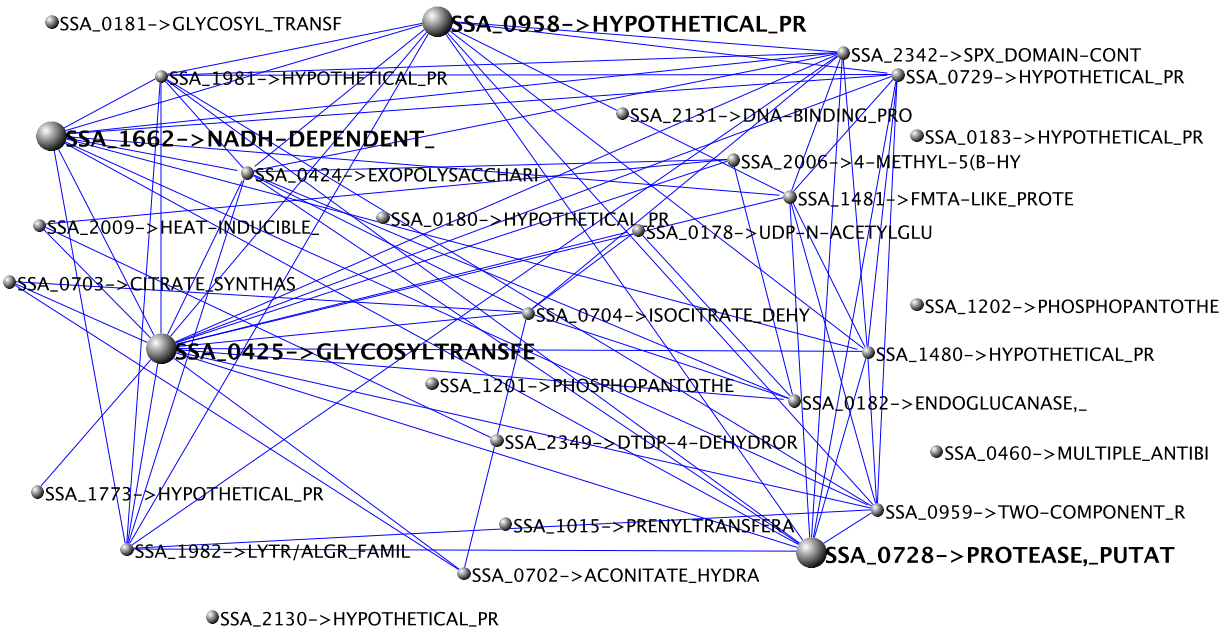

Figure 11. Network of the 30 most highly connected genes in the blue module. Connections displayed correspond to a topological overlap greater than 0.18. The top 4 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 11 connections in this subset.

	IMConnectivity	Genes
1	34.22	SSA_1261->ribose-5-phosphate_isomerase_A
2	30.60	SSA_1260->phosphopentomutase
3	28.04	SSA_1256->NAD-dependent_deacetylase
4	25.86	SSA_1037->cytidine_deaminase
5	25.63	SSA_1038->putative_lipoprotein
6	25.51	SSA_0524->microcompartment_protein,_putative
7	24.99	SSA_1919->phosphotransferase_system,_mannose-specific_EIIC,_putative
8	24.37	SSA_0525->microcompartment_protein,_putative
9	24.14	SSA_1258->purine_nucleoside_phosphorylase
10	23.96	SSA_1040->sugar_ABC_transporter,_permease_protein,_putative
11	22.82	SSA_0121->hypothetical_protein
12	22.62	SSA_1920->phosphotransferase_system,_mannose-specific_EIID,_putative
13	22.61	SSA_1259->purine_nucleoside_phosphorylase
14	22.44	SSA_2121->cell_wall_surface_anchor_family_protein,_putative
15	22.38	SSA_2111->30S_ribosomal_protein_S12
16	21.33	SSA_2109->elongation_factor_G
17	21.28	SSA_0523->aldehyde_dehydrogenase
18	21.15	SSA_0529->ethanolamine_utilization_protein,_putative
19	21.03	SSA_2262->arginyl-tRNA_synthetase
20	20.81	SSA_0522->ethanolamine_utilization_protein,_putative
21	20.80	SSA_0528->hypothetical_protein
22	20.70	SSA_1234->5'-nucleotidase,_putative
23	20.37	SSA_1946->oligopeptide_transport_system_permease_protein,_putative
24	19.91	SSA_0684->fibril-like_structure_subunit_FibA,_putative
25	19.53	SSA_2047->hypothetical_protein
26	19.17	SSA_1104->50S_ribosomal_protein_L10
27	19.06	SSA_0526->hypothetical_protein
28	19.04	SSA_0755->hypothetical_protein
29	18.92	SSA_1041->sugar_ABC_transporter,_permease_protein,_putative
30	18.84	SSA_1961->amino_acid_ABC_transporter_permease/amino_acid-binding_protein

Table 5. Top 30 most highly connected genes in the brown module. IMConnectivity is the TOM-based intramodular connectivity.

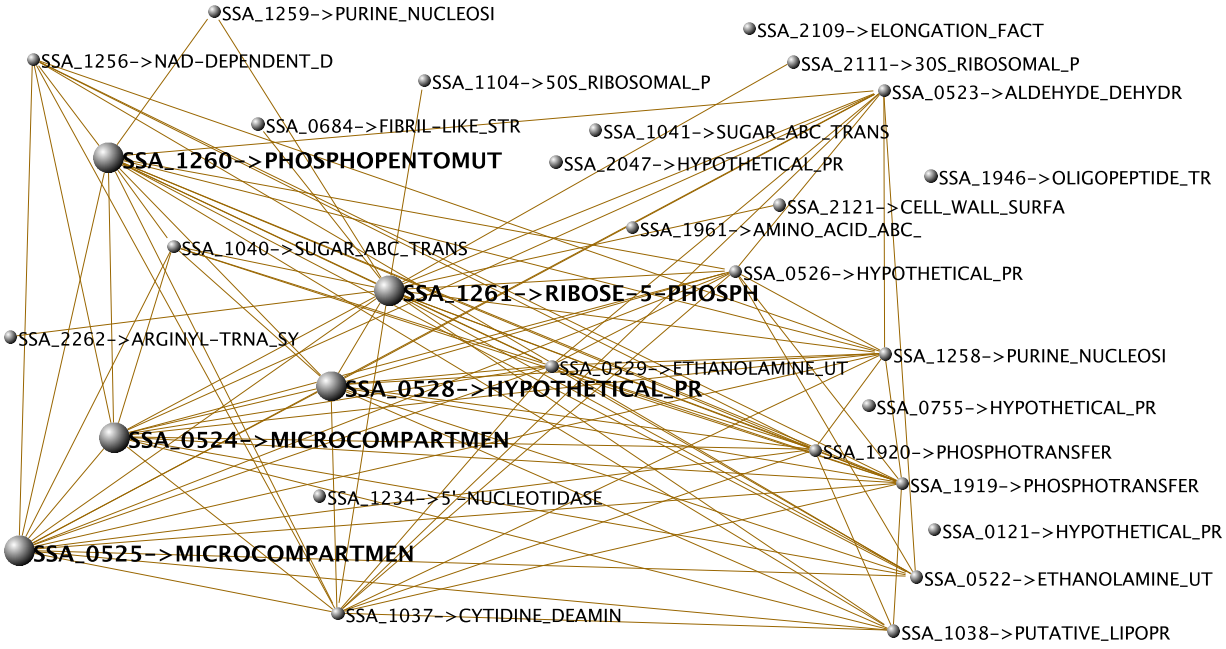

Figure 12. Network of the 30 most highly connected genes in the brown module. Connections displayed correspond to a topological overlap greater than 0.21. The top 5 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 14 connections in this subset.

	IMConnectivity	Genes
1	37.06	SSA_1591->dipeptidase,_putative
2	30.93	SSA_1695->BglG_family_transcriptional_antiterminator
3	29.88	SSA_0739->carbamate_kinase
4	26.16	SSA_0518->reactivating_factor_for_ethanolamine_ammonia_lyase
5	24.58	SSA_1008->galactokinase
6	21.53	SSA_1009->galactose-1-phosphate_uridylyltransferase
7	21.30	SSA_0342->pyruvate_formate-lyase,_putative
8	21.23	SSA_0740->C4-dicarboxylate_anaerobic_carrier,_arginine_transporter,_putative
9	20.90	SSA_1300->maltose_ABC_transporter,_permease_protein,_putative
10	20.85	SSA_0068->bifunctional_acetaldehyde-CoA/alcohol_dehydrogenase
11	20.19	SSA_1949->AliA_protein,_putative
12	20.02	SSA_0260->manganese/Zinc_ABC_transporter_substrate-binding_protein
13	19.46	SSA_0520->ethanolamine_ammonia-lyase_small_subunit
14	19.08	SSA_0738->ornithine_carbamoyltransferase
15	18.78	SSA_0777->glycogen_biosynthesis_protein_GlgD,_putative
16	18.47	SSA_0775->glycogen_branching_enzyme
17	18.36	SSA_1299->maltose/maltodextrin_ABC_transport_system,_putative
18	18.11	SSA_1010->UDP-glucose_4-epimerase,_putative
19	17.93	SSA_0262->ABC-type_Mn/Zn_transporter,_ATP-ase_component,_putative
20	17.82	SSA_1693->phosphotransferase_system_lactose-specific_component_IIBC,_putative
21	17.73	SSA_0519->ethanolamine_ammonia-lyase_large_subunit,_putative
22	17.50	SSA_0834->accessory_secretory_protein_Asp2,_putative
23	17.26	SSA_1125->NADPH-dependent_FMN_reductase,_putative
24	17.05	SSA_0737->arginine_deiminase
25	16.52	SSA_1251->HD_superfamily_hydrolase
26	16.05	SSA_1217->hypothetical_protein
27	15.87	SSA_1615->alanine_dehydrogenase,_putative
28	15.39	SSA_0071->N-acetylmannosamine-6-phosphate_2-epimerase
29	15.18	SSA_0356->dipeptidase,_putative
30	15.15	SSA_0778->glycogen_synthase

Table 6. Top 30 most highly connected genes in the yellow module. IMConnectivity is the TOM-based intramodular connectivity.

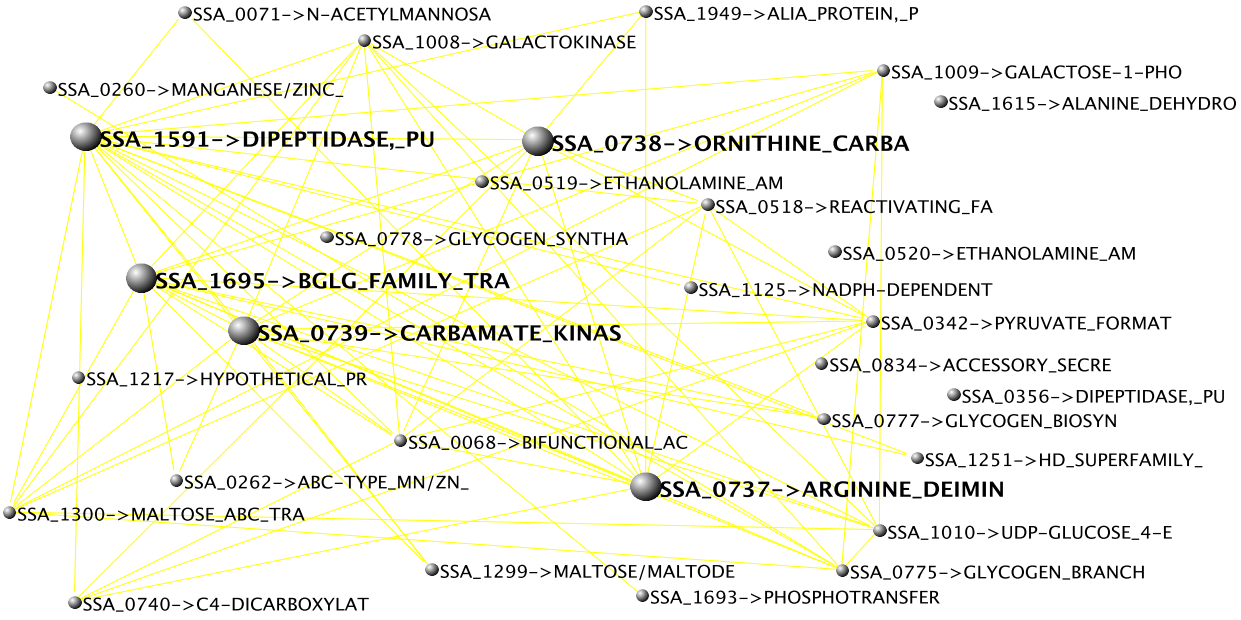

Figure 13. Network of the 30 most highly connected genes in the yellow module. Connections displayed correspond to a topological overlap greater than 0.30. The top 5 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 10 connections in this subset.

	IMConnectivity	Genes
1	13.43	SSA_1508->ABC-type_lipopolysaccharide_transport_system,_permease_component,_putative
2	12.34	SSA_0112->50S_ribosomal_protein_L22
3	11.15	SSA_1945->oligopeptide_transport_ATP-binding_protein,_putative
4	10.86	SSA_0110->50S_ribosomal_protein_L2
5	10.80	SSA_0176->DNA-directed_RNA_polymerase_subunit_beta
6	10.72	SSA_0658->hypothetical_protein
7	10.32	SSA_0109->50S_ribosomal_protein_L23
8	9.65	SSA_0115->50S_ribosomal_protein_L29
9	9.52	SSA_0107->50S_ribosomal_protein_L3
10	9.35	SSA_0108->50S_ribosomal_protein_L4
11	9.31	SSA_0111->30S_ribosomal_protein_S19
12	9.29	SSA_2191->hypothetical_protein
13	9.21	SSA_0116->30S_ribosomal_protein_S17
14	9.06	SSA_0113->30S_ribosomal_protein_S3
15	8.85	SSA_1953->NifU_family_protein,_putative
16	8.72	SSA_0106->30S_ribosomal_protein_S10
17	8.69	SSA_0655->cell_division_protein_FtsA,_putative
18	8.66	SSA_0118->50S_ribosomal_protein_L24
19	8.55	SSA_2049->polynucleotide_phosphorylase/polyadenylase
20	8.55	SSA_0352->ribonuclease_HIII
21	8.52	SSA_1507->ABC-type_lipopolysaccharide_transport_system,_ATPase_component,_putative
22	8.25	SSA_0232->hypothetical_protein
23	7.74	SSA_0657->pyridoxal_5'-phosphate_dependent_enzymes_class_III,_putative
24	7.73	SSA_1897->hypothetical_protein
25	7.71	SSA_1048->ABC_transporter_ATP-binding_protein-spermidine/putrescine_transport,_putative
26	7.59	SSA_0117->50S_ribosomal_protein_L14
27	7.50	SSA_1782->hypothetical_protein
28	7.37	SSA_0869->peptide_chain_release_factor_2
29	7.30	SSA_1779->segregation_and_condensation_protein_A
30	7.10	SSA_0870->cell_division_protein_FtsE,_putative

Table 7. Top 30 most highly connected genes in the green module. IMConnectivity is the TOM-based intramodular connectivity.

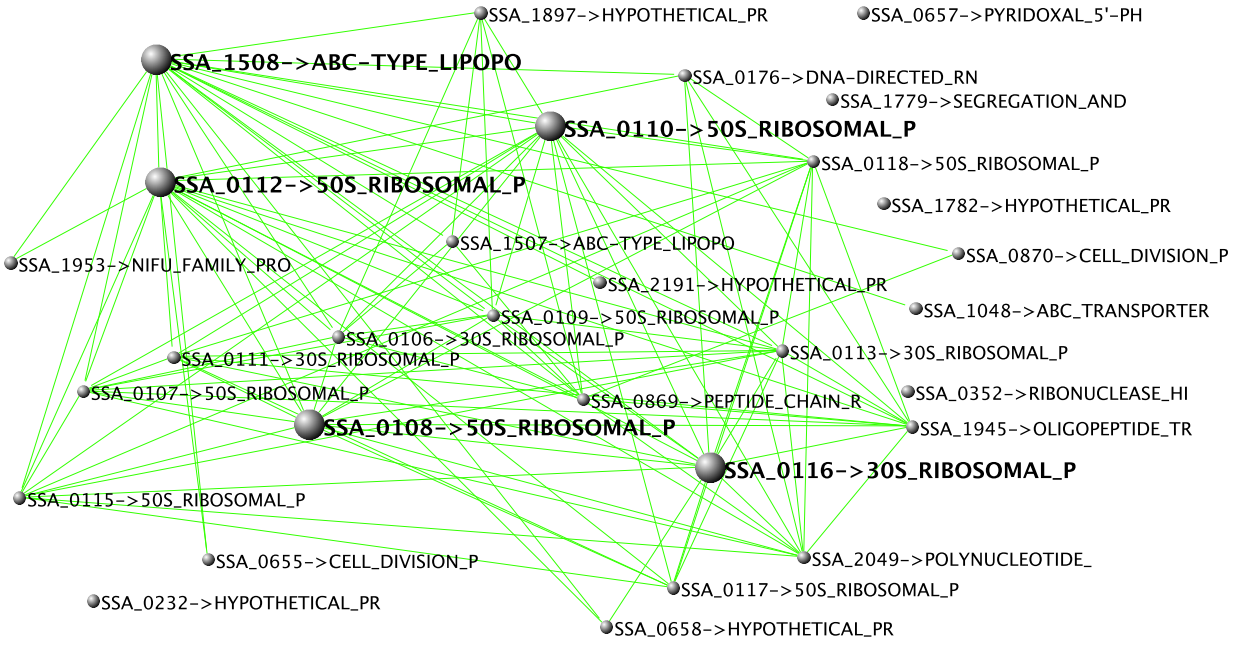

Figure 14. Network of the 30 most highly connected genes in the green module. Connections displayed correspond to a topological overlap greater than 0.11. The top 5 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 14 connections in this subset.

	IMConnectivity	Genes
1	9.34	SSA_0473->precorrin-6x_reductase,_putative
2	8.92	SSA_0485->porphobilinogen_deaminase,_putative
3	8.58	SSA_0488->glutamate-1-semialdehyde_2,1-aminotransferase,_putative
4	7.66	SSA_0491->Alpha-ribazole-5'-phosphate_phosphatase,_putative
5	7.48	SSA_0484->glutamyl-tRNA_reductase,_putative
6	6.93	SSA_0475->CbiK_protein,_putative
7	6.78	SSA_0483->siroheme_synthase,_putative
8	6.01	SSA_0477->cobalt_ABC_transporter_ATP-binding_protein
9	5.97	SSA_0496->succinylglutamate_desuccinylase/aspartoacylase_family_protein
10	5.78	SSA_0489->adenosylcobinamide_kinase
11	5.75	SSA_0478->cobalt_transport_protein_cbiN,_putative
12	5.56	SSA_0470->precorrin-4_methylase,_putative
13	5.55	SSA_0468->cobalt-precorrin-6Y_C(5)-methyltransferase
14	5.50	SSA_0472->precorrin-3B_C17-methyltransferase,_putative
15	5.49	SSA_0499->ABC-type_dipeptide_transport_system,_periplasmic_component,_putative
16	5.34	SSA_0471->cobalamin_biosynthesis_protein_CbiG
17	5.06	SSA_0486->uroporphyrinogen-III_synthase
18	5.03	SSA_0490->cobalamin_5'-phosphate_synthase,_putative
19	4.78	SSA_0476->cobalt-precorrin-2_C(20)-methyltransferase
20	4.60	SSA_0492->NADH-dependent_flavin_oxidoreductase,_putative
21	4.30	SSA_0487->delta-aminolevulinic_acid_dehydratase
22	4.26	SSA_0220->PTS_system,_mannose-specific_IIB_component,_putative
23	4.16	SSA_0221->PTS_system,_mannose-specific_IIC_component,_putative
24	3.96	SSA_0089->V-type_ATP_synthase_subunit_F
25	3.91	SSA_0495->ABC-type_oligopeptide/nickel_transport_system,_ATPase_component,_putative
26	3.71	SSA_0222->PTS_system,_mannose-specific_IID_component,_putative
27	3.61	SSA_0467->cobalt-precorrin-6A_synthase
28	2.86	SSA_0497->nickel_ABC_transporter,_putative
29	2.81	SSA_0224->hypothetical_protein
30	2.71	SSA_0088->V-type_sodium_ATPase,_subunit_C,_putative

Table 8. Top 30 most highly connected genes in the red module. IMConnectivity is the TOM-based intramodular connectivity.

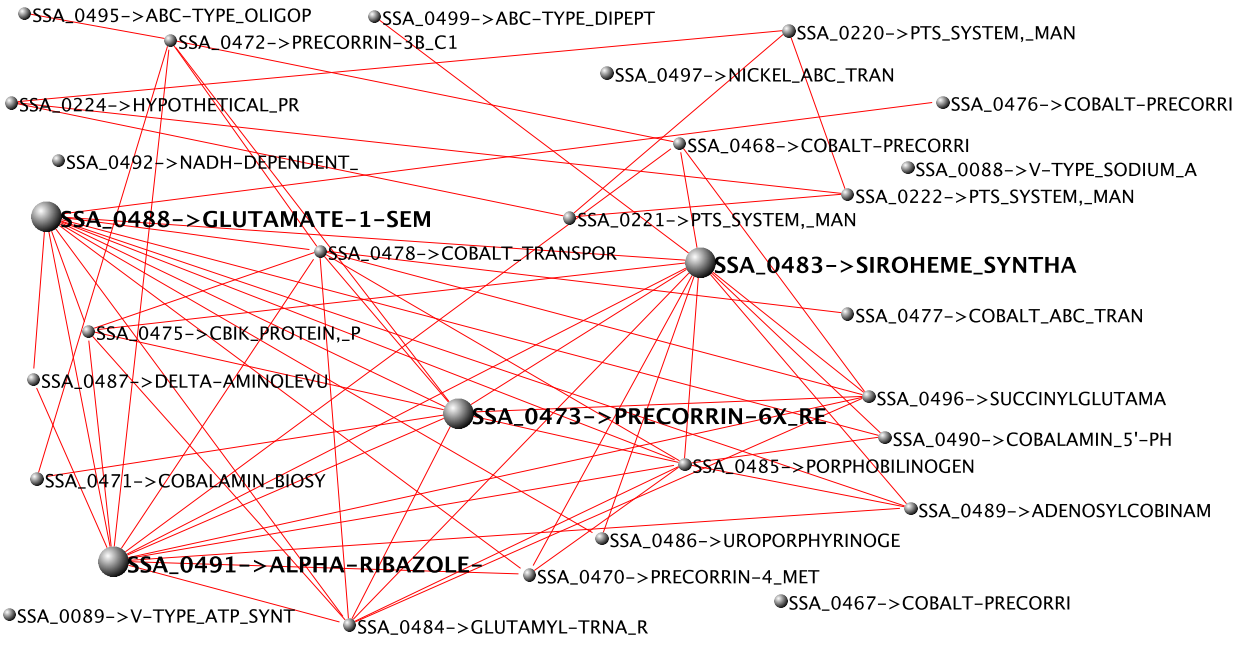

Figure 15. Network of the 30 most highly connected genes in the red module. Connections displayed correspond to a topological overlap greater than 0.11. The top 4 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 10 connections in this subset.

	IMConnectivity	Genes
1	7.74	SSA_2314->hypothetical_protein
2	6.59	SSA_1101->multidrug_resistance_efflux_pump/hemolysin_secretion_transmembrane_protein,_putative
3	6.25	SSA_1631->sortase-like_protein,_putative
4	6.22	SSA_2318->PilB-like_pili_biogenesis_ATPase,_putative
5	6.20	SSA_1632->surface_protein,_putative
6	6.08	SSA_2313->hypothetical_protein
7	5.86	SSA_2301->S-layer_protein/_peptidoglycan_endo-beta-N-acetylglucosaminidase,_putative
8	5.79	SSA_0565->hypothetical_protein
9	5.14	SSA_2307->hypothetical_protein
10	4.92	SSA_2320->hypothetical_protein
11	4.64	SSA_1447->ATP_phosphoribosyltransferase_catalytic_subunit
12	4.60	SSA_2315->hypothetical_protein
13	4.51	SSA_1444->imidazole_glycerol_phosphate_synthase_subunit_HisH
14	4.12	SSA_1100->hemolysin_exporter,_ATPase_component,_putative
15	4.04	SSA_1446->histidinol_dehydrogenase
16	3.95	SSA_1099->calcium_binding_hemolysin-like_protein,_putative
17	3.82	SSA_2303->hypothetical_protein
18	3.82	SSA_1633->FimA_fimbrial_subunit-like_protein,_putative
19	3.81	SSA_1443->1-(5-phosphoribosyl)-5-[(5-_phosphoribosylamino)methylideneamino]_imidazole-4-carboxamide_isomerase
20	3.74	SSA_1306->Trk_transporter_NAD+_binding_protein-K+_transport,_putative
21	3.73	SSA_2305->hypothetical_protein
22	3.62	SSA_1448->ATP_phosphoribosyltransferase_regulatory_subunit,_putative
23	3.60	SSA_1307->Trk_transporter_membrane-spanning_protein-K+_transport,_putative
24	3.59	SSA_2302->Type_IV_fimbrial_biogenesis_protein,_prepilin_cysteine_protease_(C20)_PilD,_putative
25	3.53	SSA_1308->hypothetical_protein
26	3.37	SSA_2304->hypothetical_protein
27	3.33	SSA_2299->hypothetical_protein
28	3.28	SSA_1439->hypothetical_protein
29	3.22	SSA_2300->hypothetical_protein
30	3.05	SSA_1634->Heme_utilization/adhesion_exoprotein,_putative

Table 9. Top 30 most highly connected genes in the black module. IMConnectivity is the TOM-based intramodular connectivity.

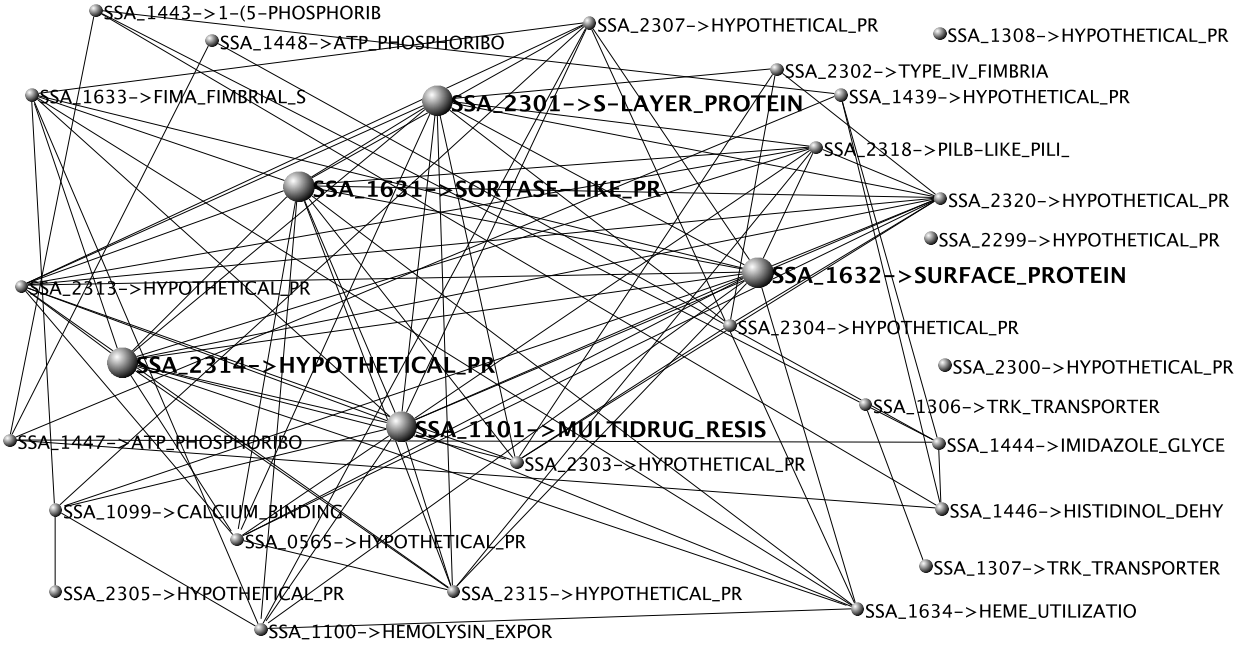

Figure 16. Network of the 30 most highly connected genes in the black module. Connections displayed correspond to a topological overlap greater than 0.11. The top 5 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 12 connections in this subset.

	IMConnectivity	Genes
1	15.02	SSA_0676->farnesyl_diphosphate_synthase,_putative
2	14.05	SSA_0675->exodeoxyribonuclease_VII_small_subunit
3	12.22	SSA_2159->hypothetical_protein
4	12.20	SSA_2245->recombinase_A
5	11.73	SSA_0677->rRNA_methylase,_putative
6	10.80	SSA_0678->ArgR_family_transcriptional_regulator
7	10.50	SSA_0679->DNA_repair_and_genetic_recombination,_putative
8	9.78	SSA_2158->methyltransferase,_putative
9	9.31	SSA_2157->DNA_repair_protein_RadA
10	8.83	SSA_0192->acetate_kinase
11	8.23	SSA_1055->hypothetical_protein
12	8.08	SSA_1717->modification_methylase_DpnIIB,_putative
13	7.16	SSA_2160->deoxyuridine_5'-triphosphate_nucleotidohydrolase
14	6.75	SSA_2370->zinc-dependent_peptidase
15	6.40	SSA_1184->DNA_topoisomerase_I
16	6.34	SSA_1183->hypothetical_protein
17	5.55	SSA_1182->tRNA_(uracil-5-)-methyltransferase_Gid
18	5.10	SSA_1716->restriction_endonuclease_SsuRB,_putative
19	5.10	SSA_0215->periplasmic_sugar-binding_protein_(ribose_porter),_putative
20	4.84	SSA_2117->DNA_recombination_protein_RmuC,_putative
21	4.47	SSA_0680->Serine/threonine_protein_phosphatase,_putative
22	4.29	SSA_2369->hypothetical_protein
23	4.25	SSA_2367->cobalt_transporter_ATP-binding_subunit
24	3.86	SSA_1210->GTP_pyrophosphokinase,_putative
25	3.83	SSA_0715->DNA_uptake_protein,_putative
26	3.69	SSA_2116->CMP-binding-factor_1, putative
27	3.61	SSA_0718->hypothetical_protein
28	3.60	SSA_1494->UDP-N-acetylglucosamine_1-carboxyvinyltransferase
29	3.49	SSA_1747->hypothetical_protein
30	3.20	SSA_0355->DNA_mismatch_repair_protein,_putative

Table 10. Top 30 most highly connected genes in the pink module. IMConnectivity is the TOM-based intramodular connectivity.

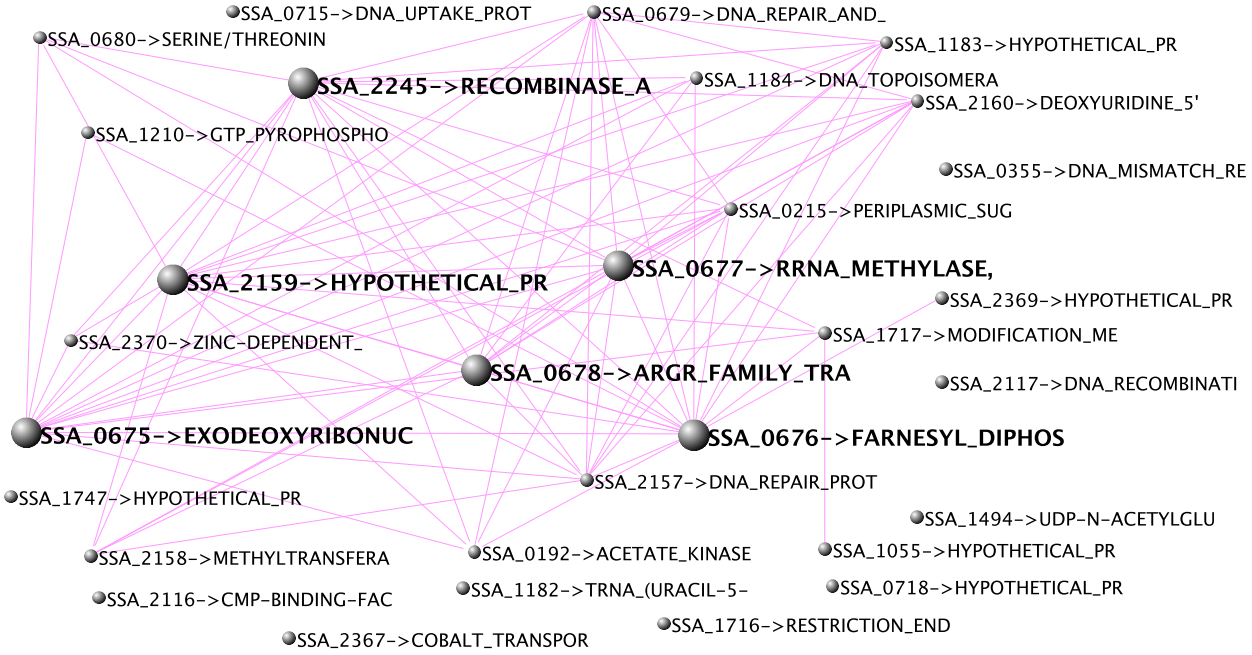

Figure 17. Network of the 30 most highly connected genes in the pink module. Connections displayed correspond to a topological overlap greater than 0.30. The top 6 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 12 connections in this subset.

	IMConnectivity	Genes
1	12.17	SSA_1649->hypothetical_protein
2	10.60	SSA_0415->permease,_putative
3	9.20	SSA_0644->DNA_protection_system,_DNA-binding_ferritin-like_protein_(oxidative_damage_protectant),_putative
4	7.54	SSA_1422->hypothetical_protein
5	7.32	SSA_2253->3-methyladenine_DNA_glycosylase_I, constitutive,_putative
6	7.13	SSA_0896->two-component_response_transcriptional_regulator
7	7.10	SSA_0846->DNA_polymerase_III_DnaE
8	6.71	SSA_0610->LemA-like_protein,_putative
9	6.38	SSA_2101->amino_acid_ABC_transporter_periplasmic_amino_acid-binding_protein
10	6.04	SSA_0459->hypothetical_protein
11	5.46	SSA_1829->RNA_methyltransferase,_putative
12	5.35	SSA_0143->hypothetical_protein
13	4.63	SSA_0212->phenylalanyl-tRNA_synthetase,_beta_subunit,_putative
14	3.83	SSA_1964->hypothetical_protein
15	3.75	SSA_2277->DNA_segregation_ATPase_FtsK/SpoIIIE_family_protein,_putative
16	3.47	SSA_1689->hypothetical_protein
17	3.02	SSA_2241->hypothetical_protein
18	2.83	SSA_1076->hypothetical_protein
19	2.76	SSA_0686->Fe2+/Zn2+_uptake_regulation_protein,_putative
20	2.55	SSA_1959->undecaprenyl_pyrophosphate_phosphatase
21	2.50	SSA_2051->oligoendopeptidase,_putative
22	2.39	SSA_1939->acyl_carrier_protein
23	2.34	SSA_1531->peptide_ABC_transporter_ATPase
24	2.27	SSA_2186->bifunctional_glutamate–cysteine_ligase/glutathione_synthetase
25	1.97	SSA_2240->Holliday_junction_resolvase-like_protein
26	1.95	SSA_0440->30S_ribosomal_protein_S18
27	1.77	SSA_1305->hypothetical_protein
28	1.57	SSA_1854->hypothetical_protein
29	1.56	SSA_0910->ABC-type_multidrug_transporter,_ATPase_component,_putative
30	1.55	SSA_1979->alkaline-shock_protein,_putative

Table 11. Top 30 most highly connected genes in the magenta module. IMConnectivity is the TOM-based intramodular connectivity.

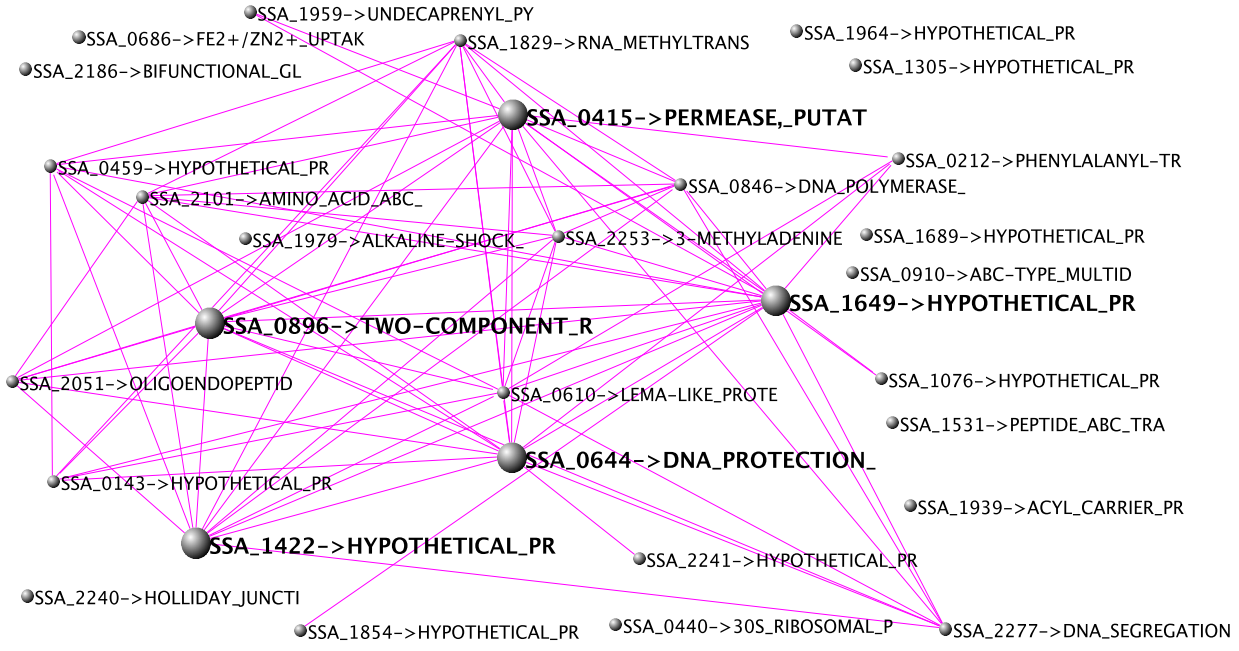

Figure 18. Network of the 30 most highly connected genes in the magenta module. Connections displayed correspond to a topological overlap greater than 0.19. The top 5 most connected genes (after topological overlap filtering) have been enlarged. Hub genes have at least 12 connections in this subset.

4. **Discussion**

A co-expression module may reveal a true biological pathway, or it may reflect noise (e.g. experimental errors, technical artifacts, or false positives) [6]. Module eigengene detection is highly robust even though noise genes (i.e. genes that truly do not belong to a given module) may be included in modules. Peripheral genes hardly affect module delineation. This is true because eigengenes are principally defined by the most highly connected intramodular hub genes [12].

Hierarchical clustering is a useful method in exploratory data analysis because it does not need *a priori* information such as specifying the number of clusters. However, cluster analysis always creates a set of clusters regardless if biologically relevant groups of genes or eigengenes actually exist. The cornerstone of gene co-expression networks is the module eigengene. Since an eigengene is defined as the first principal component, if a module eigengene is found to execute a biologically important action, then most genes in that module probably perform similarly.

Weighted gene co-expression network analysis (WGCNA) is a biologically driven data reduction technique. Similar to principal component analysis (PCA), it reduces high-dimensional data into a few meaningful groups. However, unlike PCA which requires component orthogonality between components, WGCNA allows for component interdependency. Modules may characterize biological pathways, so independence between modules cannot be assumed.

Creating an eigengene correlation network with nodes represented as eigengenes, enables the analysis of intermodule relationships. Figure 6 illustrates an eigengene network

through average linkage hierarchical clustering. The eigengene dendrogram (Figure 6) and eigengene adjacency heatmap (Figure 7) both give evidence that the some modules can be grouped into meta-modules (Module 1: Blue & Green, Module 2: Pink, Module 3: Yellow, Brown & Red, Module 4: Black, Module 5: Magenta & Turquoise). The brown and red eigengenes are highly related which is demonstrated by their low merging height in Figure 6, and also evidenced by reddish squares in Figure 7. The yellow eigengene also joins that group for similar reasons.

When constructing a network via VisANT, a TOM-based threshold for displaying connections between genes can be chosen. Depending on the degree of topological overlap, increasing the threshold can decrease the number of connections as well as the number of genes that have connections. Small threshold increases may vastly decrease the number of connections. It is important to note that the threshold can be different for every module depending on the topological overlap matrix, and the number of displayed connections needed. The initial thresholds for each module were chosen arbitrarily, and then adjusted to increase readability as well as present a similar number of top hub genes per module and a similar number of minimum connections to be considered a top hub.

Two methods (intramodular connectivity and TOM-based connectivity followed by network mapping) for identifying candidate hub genes were performed. Most modules provided similar results between the two methods (e.g. the black, pink and magenta modules). In some modules there were discrepancies between the two methods (e.g. the turquoise, and blue modules). The larger modules seemed to have the largest discrepancy between the methods, and the smaller modules had the smallest discrepancy. For these reasons, similar rankings

between the two methods can be considered equivalent and both can be used to detect candidate hub genes.

There are limitations in the analysis of this dataset. External trait information for *S. sanguinis* is not available to perform additional analyses (e.g. calculating the gene significance based on the correlation of the eigengene with an important sample trait). The resulting modules can be investigated with gene ontology information to evaluate biological significance. It is challenging to determine if modules should be individual modules or should be combined into a single meta-module. For example, the brown and the red modules are highly correlated, but it may make more sense to merge them into one module. Gene ontology information may unveil evidence that modules should be merged. The analysis of hub genes or module eigengenes may result in biologically significant pathways. Intramodular hub genes are highly correlated, therefore multiple statistically equivalent potential biomarkers are produced. These candidate biomarkers can be preferentially selected using gene ontology information. Module significance can be compared with gene significance information to determine a module of interest. Once an important module is chosen, the most highly connected genes (genes found at the tip of dendrogram branches) can be visualized (e.g. Figure 10 – Figure 18) and selected for a future study. This network analysis would help researchers create new research hypotheses and design experiments for validation of candidate hub genes in biologically significant modules.

References

[1] P. Xu, X. Ge, L. Chen, X. Wang, Y. Dou, J. Z. Xu, J. R. Patel, V. Stone, M. Trinh, K. Evans, T. Kitten, D. Bonchev and G. A. Buck, "Genome-wide essential gene identification in Streptococcus sanguinis," *Scientific Reports,* 20 October 2011.

[2] P. Xu, J. M. Alves, T. Kitten, A. Brown, Z. Chen, L. S. Ozaki, P. Manque, X. Ge, M. S. Serrano, D. Puiu, S. Hendricks, Y. Wang, M. D. Chaplin, D. Akan, S. Paik, D. L. Peterson, F. L. Macrina and G. A. Buck, "Genome of the Opportunistic Pathogen Streptococcus sanguinis," *Journal of Bateriology,* vol. 189, no. 8, pp. 3166-3175, 2007.

[3] S. Paik, S. Das, J. C. Noe, C. L. Munro and T. Kitten, "Identification of Virulence Determinants for Endocarditis in Streptococcus sanguinis by Signature-Tagged Mutagenesis," *Infection and Immunity,* vol. 73, no. 9, pp. 6064-6074, 2005.

[4] M. Trihn, X. Ge, A. Dobson, T. Kitten and C. L. Munro, "Two-Component System Response Regulators Involved in Virulence of Streptococcus".

[5] B. Zhang and S. Horvath, "A General Framework for Weighted Gene Co-Expression Network Analysis," *Statistical Applications in Genetics and Molecular Biology,* 2005.

[6] P. Langfelder and S. Horvath, "WGCNA: an R package for weighted correlation network analysis," *BMC Bioinformatics,* vol. 9, no. 559, 2008.

[7] Z. Hu, E. S. Snitkin and C. DeLisi, "VisANT: an integrative framwork for networks in systems biology," *Briefings in Bioinformatics,* vol. 9, no. 4, p. 317–325, 2009.

[8] S. L. Carter, C. M. Brechbhler, M. Griffin and A. T. Bond, "Gene co-expression network topology provides a framework for molecular characterization of cellular state," *Bioinformatics,* vol. 20, no. 14, pp. 2242-50, 2004.

[9] R. Albert and A. L. Barbasi, "Topology of evolving networks: local events and universality.," *Physical Review Letters,* vol. 85, no. 24, pp. 5234-7, 2000.

[10] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks science," *Science,* vol. 286, no. 5439, pp. 509-512, 1999.

[11] P. Langfelder, B. Zhang and S. Horvath, "Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R," *Bioinformatics,* vol. 24, no. 5, pp. 719-720, 2008.

[12] P. Langfelder and S. Horvath, "Eigengene networks for studying the relationships between co-expression modules," *BMC Systems Biology,* pp. 1-54, 2007.

Appendix: R Code

```
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")
setwd("C:/Users/edver/Desktop/Microarray Analysis")

#================================================================================

# Display the current working directory
getwd();
# If necessary, change the path below to the directory where the data files are stored.
# "." means current directory. On Windows use a forward slash / instead of the usual \.
workingDir = ".";
setwd(workingDir);
# Load the WGCNA package
library(WGCNA);
# The following setting is important, do not omit.
options(stringsAsFactors = FALSE);
#Read in the microarray data set
LeiData0 = read.csv("Lei_Micarray_Summary.csv");
# Take a quick look at what is in the data set:
dim(LeiData0);
names(LeiData0);

LeiData = LeiData0[-125,] #Remove duplicate gene, row 125

# Concatinating gene ID and gene names
LeiData$Locus <- paste(LeiData$Locus, LeiData$GeneN, sep='->')
LeiData$Locus <- gsub(" ", "_",LeiData$Locus, fixed = TRUE)

#================================================================================

datExpr0 = as.data.frame(t(LeiData[, -c(1:3)]))
names(datExpr0) = LeiData$Locus;
rownames(datExpr0) = names(LeiData)[-c(1:3)];
dim(datExpr0)

#================================================================================

# Check samples and genes for excess missing values
gsg = goodSamplesGenes(datExpr0, verbose = 3);
gsg$allOK
#Excluding 538 genes from the calculation due to too many missing samples or zero variance.

#================================================================================
```

```
if (!gsg$allOK)
{
  # Optionally, print the gene and sample names that were removed:
  if (sum(!gsg$goodGenes)>0)
    printFlush(paste("Removing genes:", paste(names(datExpr0)[!gsg$goodGenes], collapse = ", ")));
  if (sum(!gsg$goodSamples)>0)
    printFlush(paste("Removing samples:", paste(rownames(datExpr0)[!gsg$goodSamples], collapse = ",
")));
  # Remove the offending genes and samples from the data:
  datExpr0 = datExpr0[gsg$goodSamples, gsg$goodGenes]
}

dim(datExpr0)
# 42 x 1734

#=====================================================================================

sampleTree = hclust(dist(datExpr0), method = "average");
# Sample outlier detection
# sample network based on squared Euclidean distance. note that we transpose the data
A = adjacency(t(datExpr0), type = "distance")
# this calculates the whole network connectivity
k = as.numeric(apply(A, 2, sum)) - 1
# standardized connectivity
Z.k = scale(k)
# Designate samples as outlying if their Z.k value is below the threshold
thresholdZ.k = -5
outlierColor = ifelse(Z.k < thresholdZ.k, "red", "black")

datColors = data.frame(outlierC = outlierColor)
plotDendroAndColors(sampleTree, groupLabels = names(datColors), colors = datColors,main = "Sample
Dendrogram and Outlier Detection")

#=====================================================================================

# Choose a set of soft-thresholding powers
powers = c(1:20)
# Call the network topology analysis function
sft = pickSoftThreshold(datExpr0, powerVector = powers, verbose = 5, RsquaredCut=0.96)

# Plot the results:
sizeGrWindow(9, 5)
par(mfrow = c(1,2));
cex1 = 0.9;
# Scale-free topology fit index as a function of the soft-thresholding power
plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
    xlab="Soft Threshold (power)",ylab="Scale-Free Topology Model Fit,signed R^2",type="n",
    main = paste("Scale Independence"));
```

```
text(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
   labels=powers,cex=cex1,col="red");
# this line corresponds to using an R^2 cut-off of h
abline(h=0.96,col="red")
# Mean connectivity as a function of the soft-thresholding power
plot(sft$fitIndices[,1], sft$fitIndices[,5],
   xlab="Soft Threshold (power)",ylab="Mean Connectivity", type="n",
   main = paste("Mean Connectivity"))
text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers, cex=cex1,col="red")

#create Rsq table
xtable(sft$fitIndices)

sft$powerEstimate

##################  Check Scale-free topology

# here we define the adjacency matrix using soft thresholding with beta=6
ADJ1=abs(cor(datExpr0,use="p"))^6
k=softConnectivity(datE=datExpr0,power=6)

# Plot a histogram of k and a scale free topology plot
sizeGrWindow(5,5)
par(mfrow=c(1,2))
hist(k)
scaleFreePlot(k, main="Check Scale-free Topology\n")

#===================================================================================
#             One-step network construction and module detection
#===================================================================================

# power = 6, based on soft threshold calculation above

net = blockwiseModules(datExpr0, power = 6,
                #checkMissingData = FALSE,
                TOMType = "unsigned", minModuleSize = 30,
                reassignThreshold = 0, mergeCutHeight = 0.25,
                numericLabels = TRUE, pamRespectsDendro = FALSE,
                saveTOMs = TRUE,
                saveTOMFileBase = "Lei_TOM",
                verbose = 3)

table(net$colors)    # 9 Modules detected

#===================================================================================

# open a graphics window
sizeGrWindow(12, 9)
```

```
# Convert labels to colors for plotting
mergedColors = labels2colors(net$colors)
# Plot the dendrogram and the module colors underneath
plotDendroAndColors(net$dendrograms[[1]], mergedColors[net$blockGenes[[1]]],
                    "Module Colors",
                    dendroLabels = FALSE, hang = 0.03,
                    addGuide = TRUE, guideHang = 0.05)

moduleAssignments<- data.frame(sort(table(mergedColors), decreasing = TRUE))

# Module Assignments Table
xtable(moduleAssignments)

#=========================================================================================

moduleLabels = net$colors
moduleColors = labels2colors(net$colors)
# Module Eigengenes
MEs = net$MEs;
geneTree = net$dendrograms[[1]];
save(MEs, moduleLabels, moduleColors, geneTree,
    file = "Lei-networkConstruction-auto.RData")

#=========================================================================================

# Calculate topological overlap
dissTOM = 1-TOMsimilarityFromExpr(datExpr0, power = 6)
# Transform dissTOM with a power to make moderately strong connections more visible in the heatmap
plotTOM = dissTOM^6
# Set diagonal to NA for a nicer plot
diag(plotTOM) = NA

# TOM plot
TOMplot(plotTOM, geneTree, moduleColors, main = "Network Heatmap Plot, All Genes")

dim(dissTOM)
length(moduleColors)

#=========================================================================================

# Recalculate module eigengenes
MEs = moduleEigengenes(datExpr0, moduleColors)$eigengenes

MEs=orderMEs(MEs, greyLast = TRUE,
          greyName = paste(moduleColor.getMEprefix(), "grey", sep=""),
          orderBy = 1, order = NULL,
          useSets = NULL, verbose = 0, indent = 0)
```

```
# Plot the relationships among the eigengenes
sizeGrWindow(5,7.5);
par(cex = 0.9)
plotEigengeneNetworks(MEs, "", marDendro = c(0,4,1,2), marHeatmap = c(3,4,1,2), cex.lab = 0.8,
xLabelsAngle
                = 90)

#=======================================================================================

# Plot the dendrogram
sizeGrWindow(6,6);
par(mfrow=c(1, 2))
plotEigengeneNetworks(MEs, "Eigengene Dendrogram", marDendro = c(0,4,2,0),
                plotHeatmaps = FALSE)

# Plot the heatmap matrix (note: this plot will overwrite the dendrogram plot)
plotEigengeneNetworks(MEs, "Eigengene Adjacency Heatmap", marHeatmap = c(3,4,2,2),
                plotDendrograms = FALSE, xLabelsAngle = 90)

#=======================================================================================

# Truncate gene names to  fit on  VisANT plots
colnames(datExpr0)<-substring(colnames(datExpr0),1,25);

# Recalculate topological overlap
TOM = TOMsimilarityFromExpr(datExpr0, power = 6);

#=======================================================================================
#                           Export data to VisANT
#=======================================================================================

# Select module
module = "turquoise";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
modProbes = substring(modProbes,1,25);
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
                file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                weighted = TRUE,
```

```
                threshold = 0
                #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
)

# Select top 30 interconnected genes
topGenesTurquoise<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesTurquoise<-topGenesTurquoise[1:30,]
names(topGenesTurquoise)<-c("IMConnectivity","Genes")

# Select module
module = "blue";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
                file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                weighted = TRUE,
                threshold = 0
                #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
)

# Select top 30 interconnected genes
topGenesBlue<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesBlue<-topGenesBlue[1:30,]
names(topGenesBlue)<-c("IMConnectivity","Genes")

# Select module
module = "green";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
```

```
                file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                weighted = TRUE,
                threshold = 0
                #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
)

# Select top 30 interconnected genes
topGenesGreen<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesGreen<-topGenesGreen[1:30,]
names(topGenesGreen)<-c("IMConnectivity","Genes")

# Select module
module = "red";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
                file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                weighted = TRUE,
                threshold = 0
                #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
)

# Select top 30 interconnected genes
topGenesRed<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesRed<-topGenesRed[1:30,]
names(topGenesRed)<-c("IMConnectivity","Genes")

# Select module
module = "black";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
```

```
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
                        file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                        weighted = TRUE,
                        threshold = 0
                        #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
)

# Select top 30 interconnected genes
topGenesBlack<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesBlack<-topGenesBlack[1:30,]
names(topGenesBlack)<-c("IMConnectivity","Genes")

# Select module
module = "pink";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
                        file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                        weighted = TRUE,
                        threshold = 0
                        #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
)

# Select top 30 interconnected genes
topGenesPink<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesPink<-topGenesPink[1:30,]
names(topGenesPink)<-c("IMConnectivity","Genes")

# Select module
module = "magenta";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
```

```
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
                    file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                    weighted = TRUE,
                    threshold = 0
                    #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
)

# Select top 30 interconnected genes
topGenesMagenta<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesMagenta<-topGenesMagenta[1:30,]
names(topGenesMagenta)<-c("IMConnectivity","Genes")

# Select module
module = "brown";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
                    file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                    weighted = TRUE,
                    threshold = 0
                    #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
)

# Select top 30 interconnected genes
topGenesBrown<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesBrown<-topGenesBrown[1:30,]
names(topGenesMagenta)<-c("IMConnectivity","Genes")


# Select module
module = "yellow";
# Select module probes
probes = names(datExpr0)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
```

```
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
nTop = 30;
IMConn = softConnectivity(datExpr0[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
                    file = paste("VisANTInput-", module, "-top30.txt", sep=""),
                    weighted = TRUE,
                    threshold = 0
                    #probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol)
                    )

# Select top 30 interconnected genes
topGenesYellow<-data.frame(IMConn,modProbes)[order(-IMConn),]
topGenesYellow<-topGenesYellow[1:30,]
names(topGenesYellow)<-c("IMConnectivity","Genes")

rownames(topGenesRed)<-NULL
rownames(topGenesYellow)<-NULL
rownames(topGenesBrown)<-NULL
rownames(topGenesMagenta)<-NULL
rownames(topGenesPink)<-NULL
rownames(topGenesGreen)<-NULL
rownames(topGenesBlue)<-NULL
rownames(topGenesBlack)<-NULL
rownames(topGenesTurquoise)<-NULL

# Latex tables for top 30 genes
xtable(topGenesTurquoise)
xtable(topGenesBlue)
xtable(topGenesBrown)
xtable(topGenesYellow)
xtable(topGenesGreen)
xtable(topGenesRed)
xtable(topGenesBlack)
xtable(topGenesPink)
xtable(topGenesMagenta)

#=====================================================================================
#                       Intramodular Connectivity
#=====================================================================================

colorh1=moduleColors
ADJ1=abs(cor(datExpr0,use="p"))^6
Alldegrees1=intramodularConnectivity(ADJ1, colorh1)
head(Alldegrees1)
datME=moduleEigengenes(datExpr0,colorh1)$eigengenes
```

```
signif(cor(datME, use="p"), 2)
signif(datME,2)

# Calculate Module membership
datKME=signedKME(datExpr0, datME, outputColumnName="MM.")
# Display the first few rows of the data frame
head(datKME)

#=====================================================================================
# Relationship between the module membership measures (e.g. MM.turquoise)
# and intramodular connectivity
#=====================================================================================

# Plot module membership vs. intramodular connectivity by module

sizeGrWindow(8,6)
par(mfrow=c(3,3))

# For simplicity, the code is written explicitly for each module.
which.color="turquoise";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")
which.color="blue";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")
which.color="brown";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")
which.color="yellow";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")
which.color="green";
```

```
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")
which.color="red";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")
which.color="black";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")
which.color="pink";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")
which.color="magenta";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^5,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)^5")


sizeGrWindow(8,6)
par(mfrow=c(3,3))

# For simplicity, the code is written explicitly for each module.
which.color="turquoise";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)")
which.color="blue";
```

```
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)")
which.color="brown";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)")
which.color="yellow";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)")
which.color="green";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)")
which.color="red";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)")
which.color="black";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)")
which.color="pink";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
            abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
            col=which.color,
            xlab="Intramodular Connectivity",
            ylab="abs(Module Membership)")
```

```
which.color="magenta";
restrictGenes=colorh1==which.color
verboseScatterplot(Alldegrees1$kWithin[ restrictGenes],
                   abs(datKME[restrictGenes, paste("MM.", which.color, sep="")])^1,
                   col=which.color,
                   xlab="Intramodular Connectivity",
                   ylab="abs(Module Membership)")
```