CS 224W Final Report: Comparison of Community Detection Algorithms in a Gene Coexpression Network

Kai Kent - kkent17@stanford.edu - 05858825

December 19, 2016

1 Introduction

1.1 Gene Coexpression Networks

A central problem in current biology is how to identify genes and pathways that control a phenotype. Using RNA sequencing, we can get relative expression levels for each gene in a given tissue of a given subject, and a natural thing to do is to test which genes have expression levels that correlate with a phenotype. However, many species have more than 20,000 genes, necessitating a severe multiple-hypothesis correction and thus potentially missing relevant genes with moderate effects on phenotype. Furthermore, gene expression does not correlate very well with protein expression, and many genes may be involved in a phenotype without necessarily having different expression rates across phenotypic categories. To get around these problems, computational biologists often construct *gene coexpression networks* which attempt to link co-regulated genes and thus hopefully capture interrelationships among them.

One frequently use of gene coexpression networks is to partition the genes into a set of tightly-correlated modules whose union covers most of the network. Trait correlations can then be computed for 10-100 modules rather than 20,000 genes, allowing a more permissive multiple hypothesis correction and thus hopefully preserving real, moderate transcriptomic effects. In theory this also allows discovery of genes which are frequently co-regulated with significant genes, but which are not themselves statistically significant to the phenotype. This approach has led to breakthroughs in such disparate areas as evolution, disease, and neurobiology.

There are three obvious areas of study necessitated by this approach:

1. How should the gene coexpression network be constructed?,
2. How should genes be partitioned into modules?, and
3. How can modules, once identified, be evaluated and/or validated?

Many of these questions are addressed in the foundational gene coexpression network paper by Horvath and Dong [5], but we will briefly discuss them here.

1.1.1 Network Construction

Our goal in constructing a meaningful gene coexpression network is to build a weighted network where an edge's weight corresponds with the correlation in expression between the two genes it connects. Genes can be positively or negatively correlated; we can either construct a signed network to capture the type of interaction, or an unsigned network to focus only on the strength without regard for the type.

One way to construct such a network would be to create a complete graph on genes where each edge weight is exactly the correlation between the two genes. However, it doesn't make a lot of sense to draw edges between genes that are very weakly correlated, since genes that are completely uncorrelated in reality will almost always have a nonzero correlation over any finite data set. Thus we should have some minimum threshold correlation below which no edge is drawn.

The other issue to consider is that the coexpression network should be scale-free, with connectivity following a power law relationship. We infer that this should be the case by the observation that most biological networks do follow a power-law relationship. For example, the protein-protein interaction network, which is closely related biologically to gene coexpression and has well-defined links representing a binary feature (interaction or noninteraction), has a scale-free topology. Thus we should select a method of weak edge elimination and/or edge weight scaling that results in a scale-free network.

Horvath et. al. suggest to define the adjacency a_{ij} between genes i and j with coexpression similarity (correlation) s_{ij} as

$$a_{ij} = s_{ij}^{\beta}$$

for some power $\beta \geq 1$. They thusly "emphasize strong correlations and... punish weak correlations" [5]. They have empirically demonstrated the usefulness of this approach and shown that they can choose some β resulting in a scale-free network for real biological data on several biological datasets, so we took this approach to construct a weighted network.

1.1.2 Module Partitioning

As noted in [3], the module partitioning problem is *NP*-hard to solve. There exist a variety of approximate methods with good performance; we here summarize some popular approaches.

Topological Overlap Horvath and Dong suggest a hierarchical clustering procedure with a dynamic tree cutting algorithm [5]. They compute the distance between genes (nodes) using

topological overlap, then use an average linkage hierarchical clustering to construct a tree of all the nodes. Finally, they use the dynamic tree cutting algorithm described in [7].

Girvan-Newman Girvan and Newman propose in [4] to use betweenness centrality as a metric to split clusters. They briefly discuss hierarchical clustering, but note several "pathologies" such as the tendency to separate peripheral nodes from their clusters. They then propose to invert the problem: "Rather than constructing communities by adding the strongest edges to an initially empty vertex set, we construct them by progressively removing edges from the original graph." In their algorithm, they iteratively compute betweenness, remove the most-between edge, then recalculate betweenness.

1.1.3 Module Evaluation and Validation

A "good" module should be (1) biologically plausible, and (2) reasonable from a network theory and/or mathematical perspective.

All of the above methods have been designed to optimize network-level measures of module goodness, although the exact metrics do vary from method. Generally, one wants nodes in the same module to be highly connected, and nodes in different modules to be weakly connected.

The question of this project is which modularization method yields modules that have biological significance and explanatory power towards a phenotype. We planned to assess this using GO term similarity and phenotypic correlations.

GO terms The Gene Ontology (GO) vocabulary systematizes annotation of gene functions by defining standard terms and organizing them into a hierarchy [2]. There are three sets of vocabulary describing biological processes, molecular functions, and cellular components. These terms can be used to identify modules of genes that share a common pathway, function, and/or localization (and thus are biologically plausible).

Phenotypic Correlations This is a straightforward metric: given gene expression values across a set of subjects, you can see which ones correlate most tightly with a phenotype of interest. Horvath and Dong suggest that in addition to taking the average genewise correlation for a module, to pick or create a "representative" member of the cluster and compute its phenotypic correlation [5]. They suggest using the so-called "Eigengene" (the largest principal component of the correlation matrix of the genes in the module). They also experiment with using a centroid or "hub" gene with high intramodular connectivity, but found that this was less effective.

1.2 *Astatotilapia burtoni* Transcriptional Ethomics Project

Our project was to apply these gene coexpression network methodologies to data collected from an experiment in *Astatotilapia burtoni*, an African cichlid fish. *A. burtoni* males undergo rapid, dramatic changes in color and more gradual changes in morphology that correspond with their social status, either non-dominant (ND) or dominant (DOM). Males that are transitioning from a non-dominant status to a dominant one are said to be ascending (ASC). This clear, fast effect of behavioral and social inputs on physiology is mediated by transcriptional changes in the anterior pituitary of these fish, but the exact mechanisms and genes controlling the ascent process are not well understood.

We obtained data on the relative transcripts-per-million (TPM) levels of 47,807 genes in the *A. burtoni* genome in the anterior pituitary of four populations of fish: stable ND males ($n = 9$), stable DOM males ($n = 12$), ascending ASC males ($n = 9$), and females ($n = 11$), following the ascent paradigm described in [8]. We also collected behavioral data from the first half-hour of ascent for ascending fish or the first half-hour of the day for stable fish, and the last half-hour before sacrifice of all fish.

1.3 Problem Statement

We wondered which network clustering method would yield the most biologically significant results in this application. To achieve this goal, we (1) constructed a network from the transcriptional data using methods from [5]; (2) partitioned the network into modules/clusters/communities using each of the different methods in the introduction; (3) assessed the similarity of the clusters identified by the different algorithms; and (4) assessed biological plausibility and explanatory power of the clusters identified by the different algorithms.

2 Methods

2.1 Preprocessing

We wrote and ran extensive preprocessing code on the raw transcriptional dataset. We took the log transcripts-per-million (TPM) level for each gene to normalize, then threw out genes with too many zero-expression values, too little variance, or which were outliers. We also computed the expression profile correlation between each pair of subjects, and did not find any outliers. We then ran ComBat, a batch correction algorithm, to correct for different library preparation dates and sequencing runs. After filtering, we had a dataset of 7683 genes across 41 samples, with a high mean inter-subject correlation and a significant effect of experimental condition (phenotype) on

expression values.

2.2 Network Construction

We used the code published by Langfelder and Horvath [6] to select the parameter β to define the adjacencies $a_{ij} = s_{ij}^{\beta}$ as discussed in [5]. Different values of β were evaluated by the following criteria:

- Fit to a power law degree distribution (r^2 value to a linear fit on a log-log scale). We also report an r^2 value for the fit on a truncated distribution. We accept values of $r^2 \geq 0.8$.
- Slope of power law degree distribution α (so that $\mathrm{P}(k) \propto k^{-\alpha}$). Based on previous gene coexpression network analyses, we expect $1 \leq \alpha \leq 2$
- Mean connectivity (sum of a node's edge weights). We expect this to be at least 20.

After selecting the parameter β, we constructed the graph defined by the undirected (symmetric), weighted adjacency matrix A with $a_{ij} = s_{ij}^{\beta}$.

2.3 Graph Partitioning/Module Identification

We used the following publicly-available software packages for each algorithm:

- Topological overlap: WGCNA [6]
- Girvan-Newman: igraph [1]

2.4 Biological Feasibility Evaluation

2.4.1 GO term similarity

A biologically feasible module should have genes with similar gene ontology terms. Since GO terms form a hierarchical ontology, we can group GO terms together by their parents (more-general terms). To ensure that we got meaningful results, we combined related GO terms by merging them up the hierarchy until they occurred at a high enough frequency across the dataset to meaningfully assess their prevalence in each module.

To assess GO term enrichment in a module with n genes, we created a null distribution for each GO term by repeatedly selecting n random genes and counting how many times each GO term occurs. We then determined which, if any, GO terms occurred significantly more often in the module than by chance ($p < 0.05$ after Bonferroni multiple-hypothesis correction for number of GO terms). In an ideal module partition, each module would be highly enriched for a couple of conceptually related GO terms, and thus annotatable with a putative biological function.

β	r^2 fit	r^2 fit (truncated)	Power α	Mean connectivity	Network density
3	0.5987	0.9299	1.371	207.55	0.027017
4	0.7230	0.9466	1.551	96.69	0.012586
5	0.7796	0.9550	1.633	49.91	0.006496
6	**0.8185**	**0.9625**	**1.654**	**27.84**	**0.003624**
7	0.8483	0.9731	1.635	16.50	0.002148
8	0.8679	0.9762	1.615	10.27	0.001337

Table 1: Selection of weight scaling parameter β. Columns r^2 fit, Power α, and Mean connectivity are highlighted green for values in the acceptable range.

2.4.2 Phenotype correlation

We computed the "eigengene" of each module as described in [5], and computed the correlation of these eigengenes to the different social phenotypes. We did not expect that all, or even most, modules would correspond with a social phenotype. In the average case, most of the largest modules will correlate to basic cell upkeep functions. In light of this expectation, we decided to judge networks first by the **highest** magnitude of correlation of any module with the phenotype, and then by the **number of dissimilar modules** that have significant phenotypic correlations (after a Bonferroni multiple-hypothesis correction for the number of modules).

3 Results

3.1 Network Construction

As described in the Methods section, we evaluated different choices of β with results shown in Table 1. We found that $\beta = 6$ struck an appropriate balance between a good fit to the power law, a reasonable exponent α, and high mean connectivity. We then constructed a network with the adjacency matrix A with $a_{ij} = s_{ij}^{\beta}$, which as shown in the table has a power-law connectivity distribution for $\alpha = 1.654$.

3.2 Topological Overlap Clustering

We ran topological overlap clustering using the published WGCNA code [6] and mostly default parameters. We identified 10 proper modules; 820 genes were not assigned to a module. The module information is summarized in Table 2. The clustering dendrogram is shown in Figure 3, and the cluster assignments are shown in the first row.

We then computed the adjacency of each pair of eigengenes, as well as the adjacency of each to the social status of the subjects. The result is shown in Figure 1, where the adjacency between eigengenes E_I and E_J is defined as $\frac{1}{2}(1 + \mathrm{cor}(E_I, E_J))$. We also computed which phenotype had the strongest correlation (negative or positive) for each gene; the results are in Table 2. Several modules were correlated to ASC (ascenders), as well as to ND (stable non-dominant). Notably, the

Module Color	Number of genes	Strongest correlation to phenotype
unassigned (grey)	820	0.648 (female)
turquoise	1685	0.169 (female)
blue	1220	0.385 (ASC)
brown	884	0.276 (ND)
yellow	859	0.291 (ASC)
green	789	-0.190 (ASC)
red	595	-0.254 (ASC)
black	381	-0.318 (ASC)
pink	256	-0.276 (DOM)
magenta	148	-0.219 (ND)
purple	46	-0.204 (DOM)
TOTAL	**7683**	

Table 2: Module partitioning statistics for Topological Overlap clustering (WGCNA). Modules are assigned an arbitrary color, and are sorted from largest to smallest.

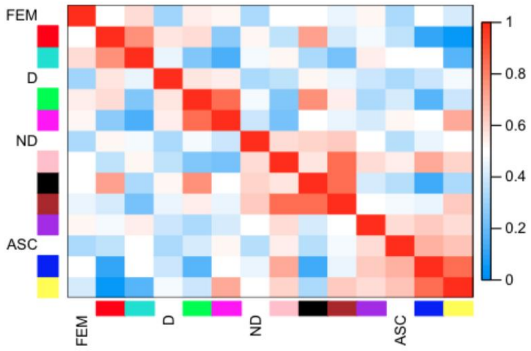

Figure 1: Eigengene adjacency heatmap for topological overlap clustering. Blue (0) represents a correlation of -1, while red (1) represents a correlation of 1.

strongest correlation found was the set of unassigned genes to the female phenotype; we hypothesize that this is because the female brain has sex difference in addition to transient social phenotype difference, so genes that are strongly sex-associated may have a lot of noise from the males and thus have unusual graph connections, making them difficult to assign to modules.

3.3 Girvan-Newman Clustering

We had many problems running this algorithm on the graph. Since the graph is complete, albeit with weak connections, computing the weighted betweenness of each edge is extraordinarily time intensive, and one would need to remove at minimum 7683 edges, recomputing betweenness each time, in order to disconnect the graph. We estimated based on small runs of the algorithm that a run on the complete data would take decades on the hardware we were using.

We tried to eliminate weak edges and smooth out variation by rounding each edge weight to the nearest 1/10th; this gave us an average degree of 138.7 and an estimated run time of approximately 3.4 years. We next tried rounding to the nearest fifth (would take about a month) and to the nearest half (would take about 8 hours). At this point, we had reduced ourselves to an

Module Color	Number of genes	Strongest correlation to phenotype
unassigned (grey)	2193	0.342 (ASC)
turquoise	350	-0.224 (ND)
brown	231	0.322 (ASC)
green	181	-0.178 (ASC)
blue	180	0.161 (ASC)
yellow	40	-0.315 (DOM)
black	14	0.266 (ND)
red	11	-0.152 (female)
TOTAL	**3200**	

Table 3: Module partitioning statistics for Girvan-Newman clustering. Modules are assigned an arbitrary color, and are sorted from largest to smallest.

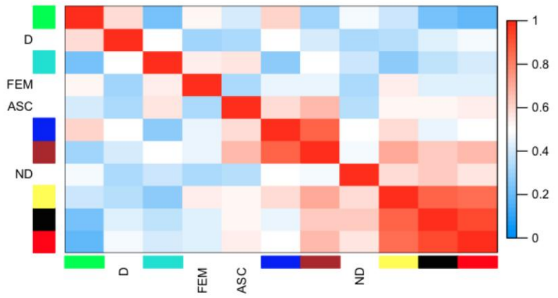

Figure 2: Eigengene adjacency heatmap for Girvan-Newman clustering. Blue (0) represents a correlation of -1, while red (1) represents a correlation of 1.

essentially unweighted graph; the remaining edge weights were identically 0.5.

We decided that if we were going to completely throw out edges with weights less than 0.25, we should at least preserve the edge weight variation in heavier edges. We ended up zeroing out all the edges with weights less than 0.15, to obtain an average node degree of 14.5. We predicted that this would take around 2 days, so we picked a random sample of roughly half of the genes (3200) and ran the Girvan-Newman algorithm on those only. We reasoned that this should preserve the strongest communities because they should be characterized by many strong connections, making it less probable that a gene would be separated from its community than that it would be separated from a neighbor not in its community. Removal of genes from dense, large communities should similarly not systematically impact the community structure, as long as the removals are random.

We identified 7 modules with at least ten genes assigned to them; the other 2193 genes we left unassigned. The module information is summarized in Table 3. The module assignments are also shown under the dendrogram from the topological overlap clustering (second row of Figure 3). As for topological overlap, we computed the adjacency of each pair of eigengenes to each other and to the phenotypes (Figure 2; also in Table 3). This partition also features several modules that are weakly correlated with non-dominant phenotype, and a few more which are correlated with ascending fish.

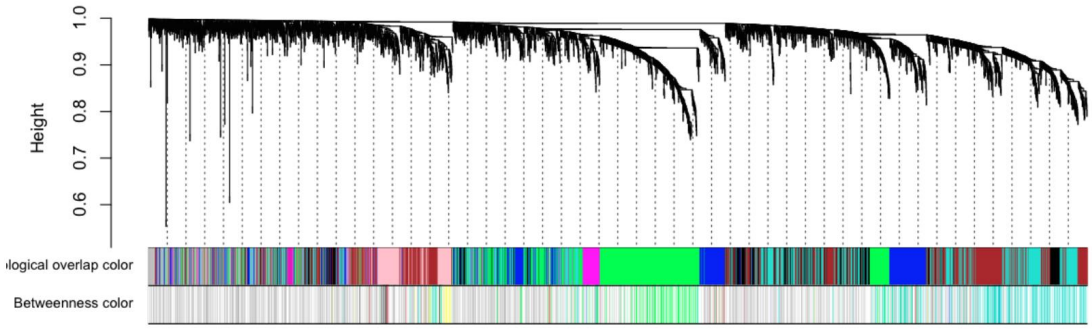

Figure 3: Topological overlap dendrogram, with module assignments at the bottom. Top row: module assignments from topological overlap clustering. Grey indicates genes that were not assigned to a module. Bottom row: module assignments from Girvan-Newman clustering. White indicates genes that were excluded for time reasons; grey indicates genes that were assigned to a module with fewer than 10 other genes.

3.4 Other Algorithms

We had initially planned to also test other community detection algorithms, but they like Girvan-Newman proved computationally intractable. We reduced the problem to a smaller size so that we could test the Girvan-Newman algorithm, but after getting our results and seeing the problems inherent to such reductions we decided it was not worthwhile to test further algorithms on a rounded-off, randomly-sampled smaller data set.

3.5 Comparison

As is clearly visible in the dendrogram in Figure 3, the module assignments from the two algorithms are fairly well preserved. The large green module from the topological overlap clustering is highly similar to the green module from Girvan-Newman; the turquoise module from the second algorithm seems to reflect a merge of the red, turquoise, and black modules from the first.

Many of the genes that are in smaller modules in the topological overlap became unassigned in Girvan-Newman. We speculate that this is because of the procedure we used to make the Girvan-Newman algorithm computable; modules that were not dense and highly-connected could easily become split apart by this methodology. While it is unfortunate that this relic made the two partitions not directly comparable, the larger modules were in fact preserved, validating the theoretical assumption that Girvan-Newman could work as a clustering tool.

We decided it was not worth it to run a computationally expensive comparison of GO term enrichment, since the two partitions are highly similar and the Girvan-Newman algorithm used only a subsample of the data.

4 Conclusion

Although other clustering algorithms yield highly similar results on this biological dataset, topological overlap was the only one that ran in a reasonable, tractable amount of time (in fact, on a standard MacBook Pro, it took only 10 minutes). Most gene expression datasets are at least this large, and the gene coexpression network construction procedure always yields a connected graph, making it likely that the other algorithms tested would have similarly weak performance on other gene expression data. The community partitioning problem is NP hard, but in this application the hierarchical bottom-up clustering procedure is ideal not because it produces better results than other algorithms, but rather because it is able to avoid computing anything costly like betweenness, much less re-computing it, and thus is able to be orders of magnitude more efficient.

References

[1] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[2] M. A. et. al. Gene Ontology: a tool for the unification of biology. *Nature Genetics*, 25, 2000.

[3] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. In *Technical Report NEC*, 2004.

[4] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences of the United States of America*, 2001.

[5] S. Horvath and J. Dong. Geometric interpretation of gene coexpression network analysis. *PLOS Computational Biology*, 4(8), August 2008.

[6] P. Langfelder and S. Horvath. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9(559), December 2008.

[7] P. Langfelder, B. Zhang, and S. Horvath. Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. *BMC Bioinformatics*, 24(5), November 2007.

[8] K. P. Maruska and R. D. Fernald. Social regulation of gene expression in the African cichlid fish Astatotilapia burtoni. In *Handbook of molecular psychology*, pages 52–78. Oxford UP, 2014.