Mining sub-graphs from gene expression partial correlations

Janet Matsen

November 19, 2016

1 Introduction

My lab studies interactions between methane oxidizing microbes and other bacteria, using microbe-rich sediment at the bottom of Lake Washington as a model ecosystem. Understanding how these communities oxidize methane would provide insight into a major greenhouse gas mitigation system. When a scoop of lake sediment is incubated with methane as the only carbon source, three types of organisms appear: (a) methane-consuming "methanotrophic" bacteria, (b) methanol-consuming non-methanotrophic "methylotrophic" bacteria, and (c) bacteria that can consume only multi-carbon compounds, called "heterotrophs". Initial steps will focus on untangling interactions between the methanotrophs, and the non-methanotrophic methylotrophs. Should the methods prove fruitful, they can be extended to the heterotrophs.

Gene expression data from 88 complex communities was previously assembled and used to produce a partial correlation matrix. I aim to use data mining and a graph database to identify sub-graph motifs that could elucidate the metabolic interactions between microbes in these samples. In particular, identification of hubs and cycles that include genes from two bacteria can be used to discover metabolic cooperation, such as nutrient exchange.

2 Approach

I am modeling the partial correlation network in Neo4j, the leading graph database. Nodes represent genes, and edges connect genes with statistically significant partial correlation with one another. The nodes are labeled with the species, gene id, and protein name. Edges are labeled with the partial correlation, and sign (positive or negative). Two gene's nodes have an edge with a large positive value when the expression of the two genes are correlated, after accounting for the expression of the rest of the genes. Conversely, large negative edge weights indicate that the corresponding genes tend to be anticorrelated after controlling for the other genes' expression.

First, I will find connected components. These will be filtered based on the most essential feature: a mixture of organism types including one methanotroph and one methylotroph. If these connected components are very large, I will seek to break them apart by filtering for magnitude of covariance, and look for cliques within. Then I will use techniques in from social network analysis to do community detection.

I will leverage existing packages when available, and write my own algorithms when necessary.

3 Related Work

Graphs are a useful tool for describing how species interact [2]. Partial correlations are often used to determine the connections in a graph in which the edges represent influence of one node on another [5]. Once a graph is in hand, we can find community structure: groups of nodes where the nodes within a group are highly connected to each other but there are fewer connections between groups [4]. Recent work has applied partial correlation to the "phyllosphere" microbiome, the collection of bacteria that live on plants [1].

4 Progress to Date

I first prepared my data for import into Neo4j. Having zero experience with Neo4j or graph databases, I first sought to load a small subset of my graph into Neo4j, and practice basic queries. I started with a CSV representing a two-organism sub-graph, which was lacking details like the organism and gene names. I was able to locate more descriptive data and merge it onto the network using Pandas in Python.

Loading this table into Neo4j required significant understanding of Cypher, Neo4j's query language. In short, data loading involves iterating over a data file, adding nodes and edges as you go. In Neo4j, it is easier to iterate over of a CSV that is on the internet than a local file, so I put mine on GitHub. Special care was required to ensure that each gene produced only one node. I later discovered that I will want to extract separate CSVs representing the nodes so I use the organism name as the node type.

First, I wanted to find connected components. Even for these simplest of tasks, people use external packages. I am working to use Neo4jSNA, a social network analysis package written in Java for Neo4j[1]. My only prior experience with Java (and compiled languages in general) was the SimpleDB exercise, and it has been surprisingly difficult to run Neo4j from Java.

Many of my initial attempts to use Neo4jSNA failed. After some digging in GitHub, I found out that Neo4jSNA was most recently updated for Neo4j 2.3.2, which is much older than the current version of Neo4j, which is 3.0.7. Even the latest 2-series version of Neo4j, 2.3.7, is not compatible with Neo4jSNA, but I was able to find the JARs for 2.3.2 via a link from the Chocolatey package manager. However, I now have this version of Neo4j running, and am able to send it queries and use some of the Neo4jSNA algorithms on it.

My first experiment used Neo4jSNA's label propagation [3] algorithm using the two-organism subgraph. I was able to use the labels produced by this algorithm and the Neo4j browser to color and visualize the clusters. To my surprise, the clusters contained many disconnected nodes. It is possible that label propagation is not a suitable algorithm for this

[1] `https://github.com/besil/Neo4jSNA`

problem. It is also possible that the graph needs to be thinned, as discussed below, before this algorithm leads to meaningful clusters.

5 Plan for Quarter

The primary goal is to find sub-graphs with nodes (genes) from more than one organism. My next step is to run Neo4jSNA's label propagation algorithm on a reduction of the two-organism subgraph comprising only edges that connect the two organisms. I will also explore the other algorithms included, with particular focus on results that include nodes from multiple organisms.

I also plan to explore further thinning of the graph. One approach is to use the magnitude of partial correlations for known interactions between these species. For the pair of organisms in my two-organism sub-graph, the methanol production/consumption genes in each species should have significant partial correlation. I will use the magnitudes of these special edges to further thin the two-organism graph. Then I can look for connected components and perform label propagation on this smaller network. I will check to make sure that the output of the algorithms is consistent with my biological knowledge.

After these tests on the two-organism sub-graph, I will scale up to the full data set. This is likely to require Amazon web services, which will require me to learn how to work on the cloud. Hopefully the jars I am using on my mac will work in AWS once I install a Java software development kit.

Third, if I have time, I will try to implement the community detection algorithm described in [4]. This algorithm works to find edges that are least central to communities in the graph and removes them. Such "between" edges can be found by looking for edges that are often used in shortest paths between pairs of nodes.

In the end, I will present the discovered sub-graphs to my lab and we will design experiments to test the proposed functional relationships.

References

[1] Matthew T Agler, Jonas Ruhe, Samuel Kroll, Constanze Morhenn, Sang-Tae Kim, Detlef Weigel, and Eric M Kemen. Microbial hub taxa link host and abiotic factors to plant microbiome variation. *PLoS Biol*, 14(1):e1002352, 2016.

[2] Ana I Borthagaray, Matías Arim, and Pablo A Marquet. Inferring species roles in metacommunity structure from species co-occurrence networks. *Proceedings of the Royal Society of London B: Biological Sciences*, 281(1792):20141425, 2014.

[3] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[4] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[5] Alfred Hero and Bala Rajaratnam. Hub discovery in partial correlation graphs. *IEEE Transactions on Information Theory*, 58(9):6064–6078, 2012.