

Data-Driven Analytics for High-Throughput Biological Applications

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Carissa Robyn Bleker

May 2020

© by Carissa Robyn Bleker, 2020
All Rights Reserved.

Acknowledgments

I would like to thank my advisor Dr. Michael Langston for the years of patient guidance and support, as well as my dissertation committee: Dr. Nina Fefferman, Dr. Audris Mockus, and Dr. Russell Zaretski.

Dr. Daniel Jacobson for introducing me to the world of Computational Biology and giving me opportunities I could hardly have imagined six years ago.

Dr. Dave Ussery for support and advice.

My fellow graduate students and treasured friends: Piet Jones, Armin Geiger, Debbie Weighill, Stephen Grady, Allan Lu, Ron Hagan, Charles Phillips and others, many of whom have served as friend, sounding board, mentor, and therapist in the past five years.

I appreciate the support of my friends and family throughout my academic career.

And finally, my husband Edward, you know why.

Abstract

High dimensional and complex biological data continues to burgeon, making the development and automation of data-driven algorithms and workflows ever-more important. Focusing on graph theoretical methods, we study graph construction and analytics for two foundational problems. In the first, we explore techniques for the thresholding of simple, undirected, edge-weighted biological graphs. In the second, we build resting state brain graphs from magnetoencephalographic data, on which we use a number of graph metrics to compare individuals, brainwaves and epoch lengths. In a separate effort, we move down the evolutionary ladder and take a look at the functional and metabolic differences between *Escherichia coli* phylotypes. Throughout, we develop novel data-driven methodologies and focus on exposing underlying assumptions of previous data-analysis workflows.

Contents

1	Introduction	1
2	Thresholding of Biological Graphs	4
2.1	Background	5
2.1.1	Data	6
2.1.2	Correlation Metrics	8
2.2	Review of thresholding approaches	8
2.2.1	Graph structure approaches	10
2.2.2	Statistical approaches	17
2.2.3	Other approaches	20
2.3	Methods	21
2.3.1	Software implementation	21
2.3.2	Software usage	23
2.3.3	Benchmarking	27
2.4	Results and discussion	28
2.5	Conclusion	38
2.6	Acknowledgements	41
	Bibliography	42
	Appendices 2	48
2.A	Benchmark graphs	49
3	Magnetoencephalographic Data Analysis: New Tools for Interpreting Neuromagnetic Profiles of Resting Brain Networks in Pre-adolescent Children	59
3.1	Introduction	60

3.2	Methods	63
3.3	Results and discussion	68
3.4	Conclusion	74
3.5	Acknowledgements	77
	Bibliography	78
4	Metabolic Comparison of <i>Escherichia coli</i> Phylotypes	82
4.1	Introduction	83
4.1.1	Historic <i>E. coli</i> population structure	83
4.1.2	Recent developments in <i>E. coli</i> phylogeny	84
4.1.3	Similar work in <i>E. coli</i> phylogenetic metabolism	85
4.1.4	Aims and approach	87
4.2	Methods	88
4.2.1	Data acquisition	88
4.2.2	Core genome pipeline	89
4.2.3	Annotation of core genomes	91
4.3	Results	94
4.4	Discussion	97
4.5	Conclusion	101
4.6	Acknowledgements	102
	Bibliography	105
	Appendices 4	110
4.A	Biocyc relabelling procedure	111
4.B	Currency compounds	112
4.C	Sample host disease per phylotype	113
5	Overall Conclusions and Directions for Future Work	118
	Bibliography	121
Vita		123

List of Tables

2.1	Metrics of association between two variables $\mathbf{x} = (x_1, x_2, \dots, x_n)$ and $\mathbf{y} = (y_1, y_2, \dots, y_n)$. $\bar{x}$ and $\bar{y}$ are the means of $\mathbf{x}$ and $\mathbf{y}$	9
2.2	Types of statistical errors.	17
2.3	Thresholding methods implemented in the software package. Established methods have previously been proposed as general thresholding techniques, revised methods are established methods that have been adapted for more general weighted graphs or based on new knowledge, proposed methods are novel techniques developed here.	22
2.4	Total number of significant GO terms per threshold in the human cell cycle graph. Note that the threshold value in the last row refers to the average per vertex threshold corresponding to the local-global alpha of 2.60.	32
2.5	Results of GO-cluster analysis per threshold method.	36
2.A.1	Graphs used in thresholding benchmarks.	50
3.1	Band pass filters	65
4.1	Number of member genomes selected from NCBI per phylotype.	90
4.2	Core genome sizes based on BioCyc genomes	94
4.3	Core genome sizes based on NCBI genomes.	95
4.4	Core genome annotations per phylotype.	98
4.B.1	List of currency compounds	112
4.C.1	<i>E. coli</i> genome sample associated host diseases.	113

List of Figures

1.1	A small example graph. Vertices are represented by circles, and edges as lines between vertices.	3
2.1	Common sources of biological data from which we can make simple correlation graphs. Such datasets span the omics spectrum.	7
2.2	Example histogram of correlations for human cell cycle graph.	24
2.3	Analysis of thresholding results as viewed in an interactive Jupyter notebook. . .	26
2.4	Calculation of thresholds. Detailed explanation in text.	30
2.5	Thresholds identified for the human cell cycle graph, as they relate to both vertex and edge count.	31
2.6	The relationship between the second eigenvalue and density in the human cell cycle graph.	31
2.7	Distributions of threshold methods across the groups of graphs. (A) Intact, (B) GDC methylation, (C) ManyMicroarray, and (D) Entropy.	34
2.8	Boxplots of the number of significant clusters per data source for methods with significant differences across data sources. Lines point to data pairs with a significant difference.	37
2.9	Relationships between the GO enrichment results and graph edge weight distribution measures for a select number of thresholding methods.	39
2.10	Precision, recall and F_1 scores of global threshold techniques in the ManyMicroarray dataset. Red asterisks denote the distribution of the score for the technique is significantly higher compared to the alternative methods.	40
3.1	MEG sensors arranged in a helmet around the head are able to measure minute changes in the magnetic field around the brain. Figure from [7].	61

3.2	WPLI distributions for different epoch lengths.	65
3.3	wPLI distributions for different epoch lengths. Each horizontal line represents a sensor, and the beginning and end ranges of the weights of its adjacent edges. . .	67
3.4	Vertex (sensor) classification according to scalp position.	69
3.5	Average local efficiency per epoch length and bandpass.	70
3.6	Average edge betweenness per epoch length and bandpass.	71
3.7	Euclidian based similarity between participants across graph metrics for a number of thresholds. Edge color and thickness represents the strength of association.	72
3.8	Top shared vertices across paracliques of individual MEG graphs for bandpasses and epoch lengths. X-axis is anterior (nose) to posterior, Y-axis is side to side. . .	73
3.9	Jaccard distributions per epoch, separated by bandpass.	75
3.10	Top shared edge graphs for bandpasses and epoch lengths. X-axis is anterior (nose) to posterior, Y-axis is side to side.	76
4.1	A past example of phylogenetic relationships between <i>E.coli</i> phylotypes. Based on Figure 1 of [7].	86
4.2	<i>E.coli</i> phylotype population structure according to [1].	86
4.3	Genome definitions for hypothetical strains A, B, and C.	87
4.4	Outline of method for each phylotype.	88
4.5	Analysis of the core genome and development of core genome cut-off estimate: (A) Core genome sizes across a random selection of genomes, for the hard core and for several soft core estimates. (B) Core genome sizes per phylotype, for range of cutoff percentages.	90
4.6	Core genome calculation for phylotype B1.	92
4.7	Analysis of the core genome and development of core genome cut-off estimate: (A) Estimated core genome sizes across random selections of BioCyc genomes using the new method, compared to using a soft core of 95%. (B) Comparison of core genome calculation and hard core size in BioCyc genomes.	93

4.8	Comparison of MetaCyc and EggNog protein cluster annotations. (A) Visualization of annotations per cluster for a random selection of core clusters of phylotype A (BioCyc genomes). Blue vertices represent clusters, green vertices represent MetaCyc annotations, and red vertices represent KEGG KO terms. Link thickness represents the number of proteins in the cluster that are associated with the annotation. (B) Histogram of annotation consistency for phylotype B2 (BioCyc genomes). The left histogram shows the majority of MetaCyc annotations are not consistent within annotations (a low percentage indicates protein annotations within the cluster are not in agreement), and the right histogram shows the majority of EggNog KO annotations are highly consistent within clusters. Since disagreement is unexpected and undesirable in our analysis, we proceeded with only using the NCBI genomes and EggNOG annotations.	97
4.9	Phylotype clustering based on reaction presence in the core genome. Red indicates presence, and white indicates absence. Reactions present in all phylotype cores are not shown.	99
4.10	Sample host disease per phylotype (column normalised).	103
4.11	Reactions associated with each core genome.	104

Chapter 1

Introduction

In this dissertation we are concerned with the application of data science, graphs, and graph theoretical algorithms to large and diverse biological datasets. We study the fundamental problem of creating a meaningful graph, and explore amenable applications in neuroscience and microbiology.

A graph is a mathematical construct consisting of vertices (also known as nodes) representing entities and edges denoting relationships between these entities. Figure 1.1 shows an example graph. The abstractness of a graph allows us to model any number of objects and systems - from transportation networks to genomic structures. A complete graph is one in which an edge is present between each pair of vertices. In biological applications, we often start with an initial complete graph. A fundamental process in creating a useful graph is removing insignificant edges, revealing the major structures under study. This process is explored in Chapter 2, where we tackle the problem of selecting a meaningful threshold for weighted biological graphs built from similarity metrics or correlation matrices.

In Chapters 3 and 4 we describe applications of data science to biological problems. In Chapter 3 we create brain connectivity graphs from resting state magnetoencephalography (MEG) data and apply graph theoretical techniques to assess differences in brain functional connectivity between subjects, time duration and brainwaves. In Chapter 4 we deal with diversity within and across *Escherichia coli* phylotypes. Using genomics, we build phylotype specific metabolic networks, and in comparing them, draw conclusions on the functional differences between *Escherichia coli* strains.

In Chapter 5 we summarize our efforts and discuss directions for future work.

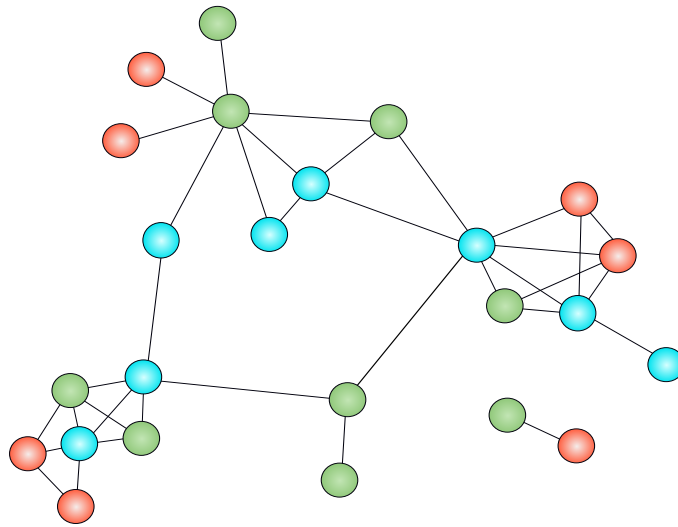


Figure 1.1: A small example graph. Vertices are represented by circles, and edges as lines between vertices.

Chapter 2

Thresholding of Biological Graphs

Abstract

In numerous biological applications, data driven graphs are produced through all-vs-all association calculations, which results in a fully connected structure. A preprocessing step before further analysis is to remove weak or irrelevant edges, and only retain those edges that correspond to strong or true signal. This process is generally referred to as “thresholding.” Two approaches to determining a graph threshold can be considered: those based on the structure of the graph, and those based on statistical modeling of the underlying data distributions. Other miscellaneous approaches have been proposed as well. Our main contributions are to modify existing approaches to fit real-world data better, introduce four novel methods, and implement a total of fifteen algorithms for comparisons. We also supply these methods in an open-source software package. Finally, we apply these thresholds to 182 biological graphs and analyze their relative solution quality. We found a method based on the largest connected component size fared best in our analyses of transcriptomic and methylation graphs.

2.1 Background

A graph is a mathematical construct consisting of a set of vertices and a set of edges connecting pairs of vertices. Vertices in general represent some objects of interest, and the edges represent association between these objects. In a directed graph the order of the pair of vertices that comprises an edge matters: one is the source, the other the target. Normally in a directed graph we refer to an edge as an arc from the source vertex to the target vertex.

In biology, data driven graphs are commonly produced through an all-vs-all similarity (or distance) calculations, resulting in a fully connected, weighted graph. These graphs can range in size from tens to tens of millions of vertices and/or edges. In order to carry out inference and perform further analysis on such a graph, we seek to remove the weakest edges that result from an all-vs-all comparison, and retain those edges that correspond to true biological signal. This process has generally been referred to as *thresholding*, but occasionally also *filtering* or *pruning*.

Many methods to determine the most applicable threshold have been proposed, but no consensus has been reached on which technique is appropriate for a particular graph. In this chapter we will compare a number of published, adapted techniques and novel techniques on both their underlying assumptions and their performance on a selection of example graphs.

Let $G = (V, E, W)$ denote a weighted graph. Let v_i be a vertex in V , and $e_{i,j}$ in E an edge between vertices v_i and v_j . Each edge $e_{i,j}$ is associated with a value $w_{i,j}$ corresponding to the strength of the association between the vertices v_i and v_j . We assume the larger $w_{i,j}$, the stronger this association. If this is not the case, we can transform the $w_{i,j}$'s to make this true (for example, if $w_{i,j}$ is a distance metric, we can use $1 - w_{i,j}$ as a similarity metric).

In this section, we will consider the process of creating such a graph from biological data, and in Section 2.2 review a number of approaches to threshold the graph.

2.1.1 Data

Figure 2.1 is a simplified illustration of datasets that are collected from biological sources. These graphs represent simple correlation graphs, in that vertices represent similar entities, such as gene-gene, transcript-transcript, or protein-protein graphs. More complicated cross-omics graphs are also possible, but beyond the scope of this dissertation.

We consider the data matrix with n rows of samples (observations) and p columns of variables (measurements):

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,p} \\ x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & x_{n,3} & \dots & x_{n,p} \end{bmatrix}.$$

While biological data comes in many forms, for the purposes of this study we consider a dataset $\mathbf{X}$ with large p and small n ($p \gg n$), from which we wish to build correlation graphs between variables. This case is probably the most common. Biological samples are often expensive, time intensive, difficult or even unethical to gather. Conversely, with modern high-throughput omics technologies, we are able to collect vast amounts of data per sample. One RNA experiment, for example, may produce measurements for thousands of transcripts, while a single DNA sample may produce millions of SNPs.

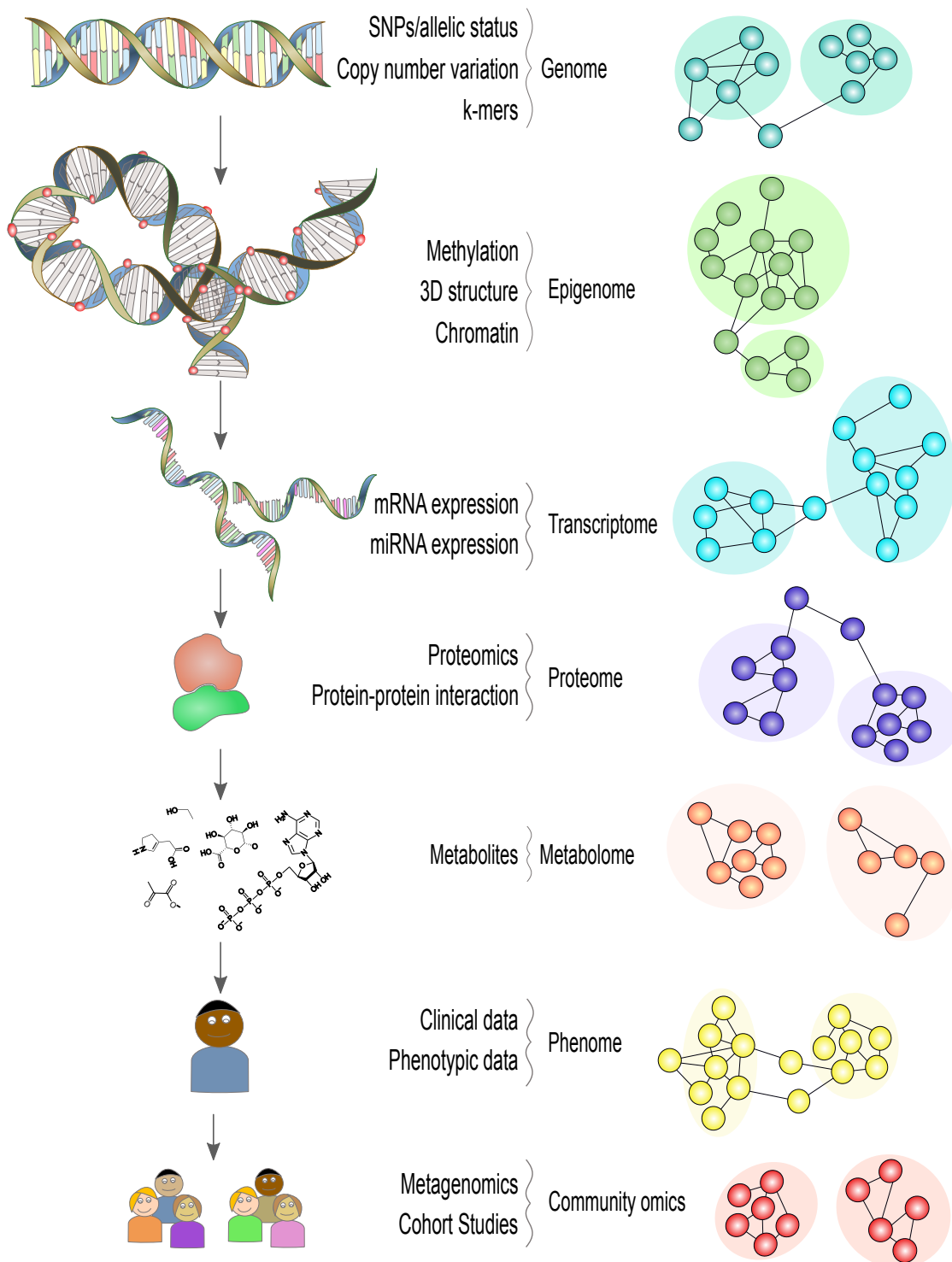


Figure 2.1: Common sources of biological data from which we can make simple correlation graphs. Such datasets span the omics spectrum.

Many statistical methods fail when the number of variables far outnumbers the number of samples, and therefore classical approaches to determine codependencies between variables cannot credibly be applied. With this in mind we will only consider thresholding methods that do not require estimation of variables.

2.1.2 Correlation Metrics

While it is not the object of this work to delve into the variety and behavior of correlation metrics, for completeness it is necessary to mention them since they are a fundamental step in graph construction. Correlation metrics are used to determine whether an association occurs between a pair of variables. The association may be linear or non-linear, and for temporal data whether the variables are in phase or not. Table 2.1 is a non-exhaustive list of common correlation metrics.

The result of an application of a correlation metric on a dataset $\mathbf{X}$, is a square, symmetric correlation matrix, which can be considered the adjacency matrix, $\mathbf{A}$, of a complete, undirected weighted graph:

$$\mathbf{A} = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \dots & w_{1,p} \\ w_{2,1} & w_{2,2} & w_{2,3} & \dots & w_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{p,1} & w_{p,2} & w_{p,3} & \dots & w_{p,p} \end{bmatrix}.$$

2.2 Review of thresholding approaches

Hard thresholding a graph with threshold t means removing edges that correspond to a weight smaller than t . It is optional then to remove vertices that become isolated as a result. When such a threshold is uniformly applied across all edges in the graph, it is called a global threshold. In contrast local thresholding is when the edges of a single vertex are considered separately from the other vertices [22]. An individual threshold is calculated for each vertex or group of vertices. Such a method considers local noise, but may also introduce local biases. For both methods, too low a threshold may result in false positives, while too high a threshold may result in false negatives.

Table 2.1: Metrics of association between two variables $\mathbf{x} = (x_1, x_2, \dots, x_n)$ and $\mathbf{y} = (y_1, y_2, \dots, y_n)$. $\bar{x}$ and $\bar{y}$ are the means of $\mathbf{x}$ and $\mathbf{y}$.

Metric	Description
Covariance	$\frac{1}{n-1} \sum_i (x_i - \bar{x})(y_i - \bar{y})$
Pearson	Covariance, normalised by the standard deviations $\frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$
Spearman	Pearson, with values are replaced by rank
Kendall rank	n_c is the number of pairs of observations where the rank of both agree, and n_d where they do not $\frac{n_c - n_d}{\binom{n}{2}}$
Czekanowski	Also know as Bray Curtis or Sorensen's quantitative similarity. $2 \frac{\sum_i \min(x_i, y_i)}{\sum_i (x_i + y_i)}$
Weighted Jaccard	For sets, Jaccard is the ratio of intersection over union. For $\mathbf{x}$ and $\mathbf{y}$ vectors, we can use the following generalization: $\frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}$
Euclidean	A distance metric that needs some transformation to become a similarity metric. $\sqrt{\ \mathbf{x} - \mathbf{y}\ }$
Cosine	$\frac{\mathbf{x}^T \mathbf{y}}{\ \mathbf{x}\ \ \mathbf{y}\ }$
Mutual Information	Given $P(x)$ and $P(y)$ are the marginal distributions, and $P(x, y)$ the joint distribution, then: $\sum_i \sum_j P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)}$

An alternative process to the global or local threshold, is soft thresholding. One soft threshold option is a function $S(G, f)$ where f is some (often sigmoidal) function applied to edge weights, that retains all edges, but weakens low edge weights and strengthens high edge weights [44]. Since soft thresholding does not remove edges, and still results in a complete graph, we will not consider it in this project.

We can consider two distinct classes of approaches to determining a graph threshold: those based on the structure of the graph, and those based on statistical modeling of the underlying data distributions. A number of miscellaneous approaches that do not fit into the previous two classes also exist.

2.2.1 Graph structure approaches

In this approach we compute a threshold value by making assumptions about the structure of the graph instead of making assumptions about individual variables and their co-dependencies. We assume that there exists an overall structure in the graph that has some contextual meaning, and attempt to find a threshold that results in the graph that best exemplifies that structure.

Scale free model The degree of a vertex is the number of vertices to which it is adjacent. In a scale free graph, the degree distribution, $p(k)$, of the vertices follows a power law distribution: $p(k) \approx k^{-\gamma}$. This means that the number of vertices with degree k is approximately $k^{-\gamma}$ where γ usually lies in the interval $[2, 3]$. Since this is an exponential function, in scale free graphs a small number of vertices are highly connected (hub vertices) and degrees of the remaining vertices exponentially decay.

Zhang and Horvath [44] argue that many biological networks empirically follow a scale free topology, and therefore a good and biologically relevant threshold can be found by fitting a scale free model to the graph.

To do this, the linear model

$$\log p(k) \sim \log k$$

is fitted to the graph for each threshold t under consideration, and the R^2 value of the model is used as a measure of how well the graph fits scale free topology. Zhang and Horvath [44] found that

a plateau is reached in the relationship between R^2 and t , and use the smallest value of t where R^2 is above 0.80 and saturation has been reached. The authors released a widely used R package called WCGNA (or Weighted Correlation Network Analysis) which is able to assist in picking this threshold [24].

More formal tests of degree distributions (chi-squared goodness of fit [23], likelihood ratio test [8]) have found that in general biological networks are not particularly scale free. Therefore, this assumption should be called into question, and only used when one is quite convinced that a scale free model fits the underlying data.

Spectral graph theory Many graphs are disconnected - they consist of two or more connected components that share no common vertices. A connected component of such a graph may consist of a number of *nearly-disconnected components* - dense subgraphs that are connected by only a few edges. Perkins and Langston [34] argue that since biological graphs often employ “guilt by association” - in a gene co-expression network vertices that share edges likely have a similar function, finding the threshold that maximizes the number of nearly disconnected components is appropriate. Equivalently this will minimize the number of bridges (edges connecting dissimilar parts) between nearly disconnected components in the graph, and separate sets of functionally similar vertices (genes).

Spectral (or algebraic) graph theory is the study of the eigenvalues of the adjacency (or the Laplacian) matrix of the graph [12]. Ding et al. [16] showed that it is possible to determine the disconnected and nearly-disconnected components using spectral methods. The number of zero valued eigenvalues of the Laplacian corresponds to the number of connected components within the graph. Additionally, if the first non-zero eigenvalue is small enough, ordering its corresponding eigenvector results in a step-like function and each step corresponds to a set of articulation points.

This resulted in the spectral methods algorithm of [34] to find the threshold maximizing the number of nearly-disconnected components:

The process in line 11 of Algorithm 2.1 to determine steps uses a moving difference of width five and flags a step when the difference is larger than $m + \frac{s}{2}$ where m is median of the differences and s the standard deviation of the differences.

Algorithm 2.1 Spectral graph thresholding [34]

```
1: procedure SPECTRALGRAPHTHRESHOLDING( $G = (V, E), t_{min}, t_{max}, \delta$ )
2:   for  $t \leftarrow t_{min}, t_{min} + \delta, t_{min} + 2\delta, \dots, t_{max}$  do
3:      $G' = H(G, t)$ 
4:      $C \leftarrow LCC(G')$  ▷ The largest connected component of  $G'$ 
5:      $L \leftarrow \mathcal{L}(C)$  ▷ The Laplacian matrix of  $C$ 
6:      $(\mathbf{v}, \mathbf{e}) \leftarrow$  eigenvalues and eigenvectors of  $L$ 
7:      $s \leftarrow \operatorname{argmin}_i \{ \mathbf{e} | e_i > 0 \}$ 
8:      $v \leftarrow v_s$ 
9:     order  $v$  smallest to largest
10:     $c_t \leftarrow 1$ 
11:    for step in  $v$  do
12:       $c_t \leftarrow c_t + 1$ 
13:    end for
14:  end for
15:   $t \leftarrow \operatorname{argmin}_{t_i} (c_{t_{min}}, \dots, c_{t_{max}})$ 
16:  return  $t$ 
17: end procedure
```

A number of improvements to this method could be considered: (1) a test to ensure the smallest non-zero eigenvalue is small enough to guarantee the existence of nearly-disconnected components in the graph, (2) either a test to ensure the largest connected component is representative of the whole graph, or count the number of nearly-disconnected components of each connected component of the graph, and (3) improve on the step detection method.

Random matrix theory Random matrix theory (RMT) was developed to analyze nuclear spectra, but has also been applied to biological networks [19, 27, 28]. RMT thresholding is based on the correlation of the eigenvalues of the graph, and is measured by the nearest neighbor spacing distribution (NNSD) of these eigenvalues. The NNSD is the distribution of the differences between subsequent (ordered) values.

A Gaussian orthogonal ensemble (GOE) is a random, symmetric matrix with independent, normally distributed entries. According to RMT for real, symmetric matrices, the NNSD of the eigenvalues of a GOE matrix has a WignerDyson distribution, and the NNSD of a non-random matrix follows a Poisson distribution [28]. Using RMT thresholding, we expect the initial correlation matrix to be random, i.e. GOE, and that the removal of low valued, or edges related to

noise will result in a non-random matrix with biological signal. To find the appropriate threshold, we iterate over t (from high to lower values), removing low weighted edges, until we find the zone of transition of the NNSD from GOE to Poisson [27]. The distributions at each point are tested using Chi square tests.

Maximal clique ratio A clique is a subgraph that contains all possible edges between its vertices. A maximal clique is a clique that cannot be enlarged by the addition of more vertices (and remain a clique). Borate et al. [6] propose a threshold based on the number of maximal cliques in the graph. Starting from the largest threshold and incrementing to the lowest the number of maximal cliques is calculated. As the number of edges increases, generally the number of maximal cliques increases. Maximal Clique-2 is the threshold before the number of maximal cliques doubles and Maximal Clique-3 is the threshold at the point before the number of cliques triples. Borate et al. [6] found Maximal Clique-2 to perform well in RNA microarray co-expression graphs.

As a generalization of Maximal Clique-2 and Maximal Clique-3, we propose the maximal clique ratio. At each increment of the threshold, we calculate the ratio of the number maximal cliques at the increment to the number of maximal cliques at the previous increment. In this case Maximal Clique-2 would be the threshold where the ratio is greater or equal to two. Aside from the increment size, another metric that can be tweaked is the minimum maximal clique size.

In order to find highly structured graphs, Palla et al. [32] uses community size. A k -clique is a clique (complete subgraph) of size k . Palla et al. [32] defines a (k -clique) community as the union of k -cliques that share $k - 1$ vertices. They choose the threshold at which the largest such community becomes at least twice as large as the second largest such community. This exhaustive bottom-up approach has since been renamed clique percolation [15]. The authors argue that this method prevents a single large community obscuring a number of smaller communities. For the selection of k (recommended to be between 3 and 6), the resultant graph should have at least half the number of original edges.

Since this method requires selecting a value for k , unrealistically requires k to be small, and is highly inefficient by design, we instead base communities on maximal cliques.

Percolation theory Percolation theory refers to the study of the flow of fluid through a porous media. In a graph, percolation is analogously related to the connectedness of the graph - if the graph is connected, information could *percolate* from one vertex of the graph to any other.

Percolation theory suggests the following thresholds:

1. Percolation threshold - Percolation is the process of randomly and independently removing edges of the graph with a probability of $1 - p$. For most random graphs there exists a critical probability, p_c , such that there exists a giant connected component orders larger than the smaller components. In reality, since we do not expect the graph to be random and are not randomly removing edges, there is likely always a giant component present in the graph.

2. Instead, we may increment the threshold, and find the point at which the size of the largest connected component of the graph has a downward inflection. This is the point at which the graph breaks apart, and information can no longer “percolate” from one part of the graph to another.

3. The highest threshold such that the graph remains connected - i.e. all vertices belong to a single connected component (incorrectly referred to as the percolation threshold in [9])

4. The highest threshold such that the graph may be split into multiple components, but is whole in the sense that no isolated vertices exist, or are removed as a result of the threshold function.

Graph density The density of a graph is the ratio of the number of edges present in the graph against the number of all possible edges:

$$d = \frac{|E|}{\frac{1}{2}|V|(|V| - 1)}.$$

Aoki et al. [2] argue that the biologically relevant threshold in a co-expression graph is the threshold that minimizes the density of the graph. More generally we may calculate density for a number of thresholds and study the behavior to see if any inflection points occur.

In [29] the threshold is found similarly in an *Arabidopsis* gene co-expression graph. The authors based their choice of threshold on minimizing the density, arguing that at the point of

minimum density, the rate of edge removal is slower than the rate of vertex removal, and the remaining edges with high weight connect a smaller number of vertices, allowing the identification of densely connected, biologically significant modules.

Global-local thresholding Guzzi et al. [22] propose a threshold method for Semantic Similarity graphs that combines local and global thresholding. For each vertex, the local threshold is:

$$t_i = \mu_i + \alpha \sigma_i,$$

where μ_i is the average weight of the edges connected to the vertex, α is a global parameter and σ_i is the standard deviation of the weights connected to the vertex. The authors use $\alpha \in [0, 1]$. For each edge, with weight $w_{i,j}$, if $w_{i,j}$ is larger than both t_i and t_j , then it is included in the graph with a new weight of 1. If $w_{i,j}$ is only larger than one of t_i and t_j , then it is included in the graph with weight 0.5. Otherwise it is not included into the graph. Vertices with degree one are removed from the final graph. If the graph contains nearly disconnected components, the process stops, otherwise the parameter α is decreased, and the process repeated.

While the authors do not show how the thresholding technique affects resulting graphs, they do show that graphs with high semantic functional consistency after thresholding also exhibit nearly disconnected components.

Clustering and edge weight distribution Apeltsin et al. [3] propose a method to threshold protein sequence similarity graphs. They define two functions of the threshold t : $Nn(t)$ the number of vertices with degree of one or greater and $SE(t)$, the number of edges present. Based on these properties, the value of interest is $Nsv(t) = SE(t)/Nn(t)$, and the ideal threshold, t^* , occurs at the smallest t where $dNsv(t)/dt > 0$.

Their argument is as follows: At low t , $SE(t)$ decreases and $Nn(t)$ remains stable, therefore $dNsv(t)/dt$ is expected to be negative. At this point the majority of edges that are removed are between protein families, and family clusters remain well connected. At t^* , outlier vertices become disconnected, and if the edge weight distribution is skewed to the left, $SE(t)$ continues to decrease, but at a lower rate. At approximately this point, $Nsv(t)$ increases, meaning $dNsv(t)/dt$ becomes positive.

Apeltsin et al. [3] found that thresholding a graph in general improved the clustering quality when compared to characterized protein families, and specifically the method they developed was able to identify the range of optimal threshold to improve clustering algorithms. This method does however require that the edge distribution is left skewed, and the number of edges rapidly decreases as the edge weight increases. They found this to be true in the protein similarity graphs they presented, but likely this would not always be the case.

Clustering coefficient Let d_i denote the degree of vertex v_i and D_i denote the number of edges between the neighbors of v_i . Then the clustering co-efficient of v_i is:

$$c_i = \frac{D_i}{\frac{1}{2}d_i(d_i - 1)},$$

the ratio of existing edges to all possible edges between the neighbors of v_i .

The clustering coefficient of the graph is the average of the c_i 's [43]:

$$C = \frac{1}{|V|} \sum_i c_i.$$

Gupta et al. [21] propose a threshold value based on observed behavior of the clustering coefficient. For a complete graph, $C = 1$, and as edges are removed it decreases. They argue that at higher thresholds, signal in the network manifests in highly connected, “cliquish” subnetworks appearing, which results in C increasing. By this logic, the critical threshold is the highest value at which a sharp increase occurs.

Following on Gupta et al. [21], Elo et al. [17] propose that, as the threshold is increased, removed edges are due to noise as long as the difference between the observed C and a randomized counterpart, C^r , is monotonically increasing. Let C and C^r be functions of the threshold t . Then the critical clustering coefficient of the graph, $C(t^*)$, occurs at the smallest threshold t , such that the difference between $C(t)$ and $C^r(t)$ is larger than the same difference of the graph at the next (incremented) threshold. The randomized counterpart, C^r , is calculated from a random network produced by the configuration model, which preserves G 's degree distribution [7].

In contrast to [21], this critical value can be calculated analytically, and accounts for random noise in the data by comparing C to C^r . When compared to predefined thresholds, and thresholds

based on significance on a number of micro-array datasets, Elo et al. [17] found that their method resulted in graphs that corresponded far better with gene ontology.

2.2.2 Statistical approaches

Statistical methods attempt to ascertain and/or assume the distribution of the data and determine significant associations based on statistical tests. A drawback to such methods is that many of them do not take into account the biological strength of the edge weights - a highly significant edge weight may have very weak biological signal and in actuality not be of interest.

Significance and power Significance and power are methods to establish the likelihood of errors when performing hypothesis tests. In the case of edge weights, we would generally be testing the following hypothesis:

$$H_0 : w = 0 \qquad H_a : w \neq 0.$$

Table 2.2 shows the two types of errors that can occur in such a statistical test.

The Type I error rate (the false positive rate) is controlled by setting the significance α (an acceptable probability of incorrectly rejecting the null hypothesis), and rejecting based on the p -value. Exact p -values are possible to calculate if we can make assumptions on the distribution of the correlations. For example, if the metric being used is Pearson Correlation Coefficient, ρ , and the null hypothesis is $H_0 : \rho = 0$, then the two-sided p -value for a measured sample correlation r_s can be calculated from a Student's t distribution with $n - 2$ degrees of freedom [13] :

$$t_s = \frac{r_s \sqrt{n-2}}{\sqrt{1-r_s^2}} \sim T_{n-2} \tag{2.1}$$

$$p\text{-value} = 2P(T_{n-2} > t_s).$$

Table 2.2: Types of statistical errors.

		Reality	
		H_0 is true	H_0 is false
Decision	Reject H_0	Type I error	Correct decision
	Fail to reject H_0	Correct decision	Type II error

Since a number of comparisons are being made, multiple hypothesis corrections may be necessary. With a given choice of significance level, we would select the edges that correspond to a p -value smaller than α . This would control the Type I error.

The critical value for $t = t_{crit}$ as given in Equation 2.1, given a significance level α is

$$t_{crit} = T_{n-2}^{-1}(1 - \alpha/2),$$

where T_{n-2}^{-1} is the inverse cumulative distribution function.

At this critical value we can calculate r_{crit} :

$$\begin{aligned} \frac{r_{crit}\sqrt{n-2}}{\sqrt{1-r_{crit}^2}} &= t_{crit}, \\ \implies \frac{r_{crit}^2(n-2)}{1-r_{crit}^2} &= t_{crit}^2, \\ \implies r_{crit}^2(t_{crit}^2 + n - 2) &= t_{crit}^2, \\ \implies r_{crit}^2 &= \frac{t_{crit}^2}{(t_{crit}^2 + n - 2)}, \\ \implies r_{crit} &= \sqrt{\frac{t_{crit}^2}{(t_{crit}^2 + n - 2)}}. \end{aligned}$$

An example use of this method is in [25], where they use Bonferroni p -value correction and control the error rate at 0.01 (while also only selecting edges weights in the highest and lowest 0.5% of values).

In contrast to Type I error, a threshold based on the Type II error (false negative rate), attempts to control the power of the test. For Pearson, we test the following hypotheses:

$$H_0 : \rho = 0 \qquad H_a : \rho = r.$$

The normal approximation of r_s , r_{crit} and r can be given by the $\arctanh$ transformation:

$$z = \arctanh(r) + \frac{r}{2(n-1)},$$

(or alternatively the Fisher transformation) [13], which applied to r_s , r_{crit} and r , gives z_s , z_{crit} and z_r respectively. The distribution of z_s under the alternative hypothesis is normal with mean z_r and standard deviation $\frac{1}{\sqrt{n-3}}$. As above, statistical power is the probability of rejecting H_0 when H_a ($\rho = r$) is true:

$$\begin{aligned}
\text{Power} &= P(\text{reject } H_0 | H_a), \\
&= P(z_s \geq z_{crit} | H_a) + P(z_s \leq -z_{crit} | H_a), \\
&= P\left(Z \geq \frac{z_{crit} - z_r}{1/\sqrt{n-3}}\right) + P\left(Z \leq \frac{-z_{crit} - z_r}{1/\sqrt{n-3}}\right), \\
&= 1 - \Phi\left(\frac{z_{crit} - z_r}{1/\sqrt{n-3}}\right) + \Phi\left(\frac{-z_{crit} - z_r}{1/\sqrt{n-3}}\right).
\end{aligned}$$

One approach to control power is introduced in [6]. For a number of values of $r = 0.99, 0.98, \dots$, the power is calculated using significance of 0.05, and the value of r when the power is at least 0.80 is used as the threshold.

When we cannot make an assumption on the distribution of the correlations, we can estimate the p -value from empirical distributions derived from permutations or bootstrapping. For a number of iterations we can either permute or sub-sample (bootstrap) the original data-matrix and recalculate the correlation. After enough iterations, the recalculated values become a background distribution that can be used to calculate the significance or power of the original correlations.

Graphical model formulation A Bayesian network is a probabilistic framework for modeling dependencies between variables based on Bayesian statistics and Directed Acyclic Graphs (DAG). The variables in the model are represented by the vertices in the graph, and edges represent the existence of conditional dependency between pairs of variables. Bayesian networks provide an analytical framework for identifying patterns of dependence/independence among variables.

Given a random vector $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$, with X_i a random variable, then the corresponding graph is $G = (V, E)$, where each $v_i \in V$ corresponds to X_i in $\mathbf{X}$ and edges between these vertices indicate the conditional distributions – each child vertex is dependent on its parent(s). The Bayesian network is $B = \langle G, \Theta \rangle$, where G is the DAG and Θ is the joint distribution.

A hurdle in this area is to derive the “true” structure of conditionality among the variables given observations, and a number of algorithms have been proposed and applied in the literature. The joint distribution is estimated through a learning process that focuses on identifying associations between variables, and the order and directionality of the associations. There are two general categories of learning algorithms: the search-and-score category, and the constraint-based category [4]. Many of these methods are *NP*-complete [10].

Scutari and Nagarajan [36] developed a thresholding method based on Bayesian networks using bootstrapping and model averaging to identify significant edges in molecular networks. In short, subsets of the original data are used to produce Bayesian networks, and these Bayesian networks averaged to separate significant from non-significant edges.

Unfortunately, Bayesian networks and learning algorithms have a number of drawbacks. Firstly, they cannot be used to model cycles (such as feedback loops). Secondly, Bayesian learning algorithms have not been found to be scalable, and thus these methods have only been applied to small networks (≤ 100 nodes/edges) for both simulated data and real-world data. In fact, “Big Data” in [37] refers to datasets with millions of observations and few variables. In general, biological dataset have thousands to millions of variables and orders fewer observations, making it harder if not impossible to apply Bayesian techniques to obtain a quality threshold.

2.2.3 Other approaches

Rank based In this approach, for each vertex, the d highest ranked adjacent edges are selected [35]. The authors recommend selecting a d between 3 and 5, which in simulated and real-world data usually resulted in a connected graph. This method, as in the global-local thresholding method, takes into account the heterogeneous aspects of biological networks, allowing a range of correlation strengths to be included in the graph.

Arbitrary value In transcriptomic networks Freeman et al. [18] use arbitrary Pearson thresholds of 0.70 and 0.90, while Zhou et al. [45] uses a cut-off of 0.60.

Top x percent Correlations are ranked by strength, and the highest x percent are retained as edges in the graph. Choosing x is an arbitrary process, based on assuming the top x percent

of edges are the only edges with biological significance. Pellegrino et al. [33] found the top 1% of co-expression correlations across cDNA micro-array datasets to be the most enriched in orthologous pairs of expressed sequence tag clones, and based on them Ala et al. [1] use the top 1% of correlations as edges in their co-expression graphs. As mentioned before, after controlling for multiple hypotheses, Lee et al. [25] select edges with weights in the top and bottom 0.5%.

Relevance networks Voy et al. [41] found that using a Pearson correlation threshold of 0.875 for a microarray dataset removed the edges that connected gene-probes to control-probes on the array, therefore removing non biologically relevant edges. In general, this approach would require knowledge of the underlying system, and selection of a threshold that supports this prior knowledge.

2.3 Methods

2.3.1 Software implementation

We developed a software package that allow users to compute statistics from the previous section and use the results to identify a practical threshold. Such an overarching collection and implementation of thresholding techniques has not, to our knowledge, previously been released.

All software is available in a repository at <https://github.com/carissableker/thresholding>. The igraph C library was extensively used for graph analysis [14], and the ALGLIB library [5] for statistical distributions. The components of the repository are:

- C++ threshold analysis code
- C++ threshold code
- Bash (and awk) threshold script
- Bash (awk and Python3) script to output a histogram of the edge weights of a graph
- Python3 Jupyter notebook to analyze the results of the thresholding analysis code

Table 2.3 lists the 15 implemented threshold techniques. Of them, four are newly devised here, two are significantly revised, and 11 have been previously proposed in cited literature. Of the fifteen as defined here, only implementations of scale-free and RMT have previously been publicly released, but here we improve the implementation of the goodness-of-fit-tests for both.

Table 2.3: Thresholding methods implemented in the software package. Established methods have previously been proposed as general thresholding techniques, revised methods are established methods that have been adapted for more general weighted graphs or based on new knowledge, proposed methods are novel techniques developed here.

Method	Novelty	Note
apeltsin	established	Implementation of the algorithm described in [3], based on the number of edges and the number of vertices.
cc-inflection	proposed	The point at which the size of largest connected component sharply inflects downwards.
density-min	established	The threshold that minimizes the density, from [2].
elo-clustering	established	Smallest threshold at which the difference between the clustering coefficient of the graph and the clustering coefficient of a random counterpart graph is larger than the same difference at the next threshold [17]
gupta-clustering	established	Threshold at which the clustering coefficient sharply increases [21]
mcr-2	established	Maximal clique 2 described in [6].
mcr-3	established	Maximal clique 3 described in [6].
mcr-max	proposed	Generalization of maximal clique 2 and maximal clique 3.
scale-free	revised	Based on WCGNA, but this method uses maximum likelihood to determine γ , and a Kolmogorov-Smirnov test to determine the goodness-of-fit.
single-component	proposed	The highest threshold that guarantees a connected graph.
spectral-methods	established	Implementation of the algorithm described in [34].
rmt	established	Implementation of the algorithm described in [19, 27, 28].
whole-graph	proposed	The highest threshold that guarantees all vertices remain in the graph.
local-global	revised	Method of [22], but adapted for general weighted graphs instead of semantic similarity graphs.
TypeI- $\langle\alpha\rangle$	established	Given any value of α significance, calculates the minimum Pearson threshold to guarantee a Type I error rate of at most α .
Power- $\langle\beta\rangle$	established	Given any value of β , calculates the minimum Pearson threshold to guarantee a Type II error rate of at most $1 - \beta$.

2.3.2 Software usage

Usage of the software components are described in the repository README file. In general, the user specifies an input graph, and options include the upper and lower bounds on thresholds to test, the threshold increment, a choice of the methods to carry out, and various parameters for the methods (including the window-size used in spectral methods and the minimum graph size). All methods can be attempted for each threshold increment. However, since certain methods are more time consuming, one can select which methods are performed.

As an example, we consider a human cell cycle graph from [11]. This graph was built from 13 time points of human gene expression data collected for 7,700 micro-array probesets. The initial complete graph has $|V| = 7,700$ and $|E| = 25,038,426$ Pearson correlations. To produce a histogram of the correlations, one can use the histogram script:

```
$ ./edgeWeightHistogram HumanCellCycle.ncol ./HumanCellCycle.edge 0.05
Edge distribution at HumanCellCycle.edge.tsv and HumanCellCycle.edge.svg
```

This produces both a tab-delimited file of the bins and bin counts, and a figure of the histogram (Figure 2.2). This histogram can be used to determine upper and lower bounds on the threshold.

To perform the analysis of possible thresholds, one can use the main analysis program. The usage is as follows:

```
Usage:
./analysis [-OPTIONS]... <GRAPH FILE PATH> <OUTPUT FILE PATH>
Graph has to be in .ncol format.
Output file path is the prefix to the results files, which will be of the form:
    <OUTPUT FILE PATH>.pid.<method_name>.txt
Options:
-l --lower          <value>    lower bound on thresholds to test (default 0.5)
-u --upper          <value>    upper bound on thresholds to test (default
    0.99)
-i --increment      <value>    threshold increment (default 0.01)
-w --windowsize     <value>    sliding window size for spectral method (
    default 5)
-p --minimumpartitionsizesize <value> minimum size of graph or subgraph after
    threshold (default 10)
-n --num_samples    <value>    number of samples for significance and power
    calculations (default NULL)
-b --bonferroni_correction      switch to perform bonferroni corrections in
    significance and power calculations (default FALSE)
-c --minimum_cliquesize <value> minimum size of maximal cliques in maximal
    clique count (default 5)
```

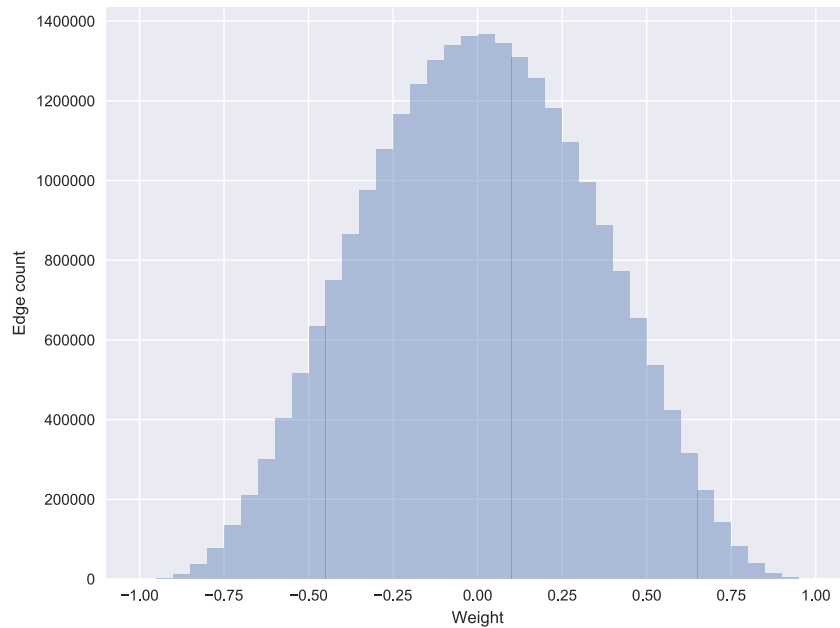


Figure 2.2: Example histogram of correlations for human cell cycle graph.

```

-m --methods      <value>    comma separated list of methods (defaults to
none)

                                0 - all
                                1 - significance and power calculations (
                                  only valid for Pearson CC)
                                2 - local-global
                                3 - scale free
                                4 - maximal cliques
                                5 - spectral methods
                                6 - random matrix theory
                                7 - clustering coefficient methods
                                8 - percolation methods

-h --help          print this help and exit

```

For our example, we can use it as follows:

```

$ ./analysis HumanCellCycle.ncol ./HumanCellCycle -l0.2 -n13 -m2,3,4,5,6,7
-----
input graph file:    HumanCellCycle.ncol
output file prefix:  HumanCellCycle.6945.
lower threshold:     0.2
upper threshold:     0.99
threshold increment: 0.01
-----
Loading graph ... done.

```

```

Number vertices: 7077
Number edges:    25038426 (maximum possible number edges 25038426)
-----

Number steps: 80
Step: 1, Threshold: 0.2 Removed 10520301 edges and 0 vertices.
Step: 2, Threshold: 0.21 Removed 481534 edges and 0 vertices.
Step: 3, Threshold: 0.22 Removed 476696 edges and 0 vertices.
.
.
.
Step: 79, Threshold: 0.99 Removed 33 edges and 13 vertices.
Done.

```

Since maximal cliques in an almost complete graph are computationally time consuming, we can use a higher lower bound for just that method:

```
$ ./analysis HumanCellCycle.ncol ./HumanCellCycle -l0.6 -m1
```

On completion, these commands will produce multiple output files, which we can combine and analyze using the Jupyter notebook. A library of custom Python functions combine the output files, supplied with either a list of output files or the output file prefix as given to the analysis program. This allows for multiple restarts in case of failures, restarts with different thresholds limits and increments, or method choices. Figure 2.3 shows a screenshot part of the results viewed in the notebook.

In the figure, each method is listed with its suggested threshold. In this example, random matrix theory did not converge, and looking closer at the individual results per threshold in the dataframe this was found to be due to the graph never leaving the GOE distribution ($p < 0.05$). The other methods have thresholds across the range of 0.70 to 0.95.

Additionally, all values collected in the threshold analysis process are readily available in dataframes for exploration at the leisure of the user.

For the final step, once the user has decided on a threshold, they can use the software to threshold the graph. For simple thresholding, the fastest is to use the `absolute_global_threshold` script. This thresholds the graph at an input value t , i.e. removes all edges with absolute value smaller or equal to t . Alternatively, the program `threshold` can also be used to threshold the graph, and has more options including local-global thresholding and rank thresholding.

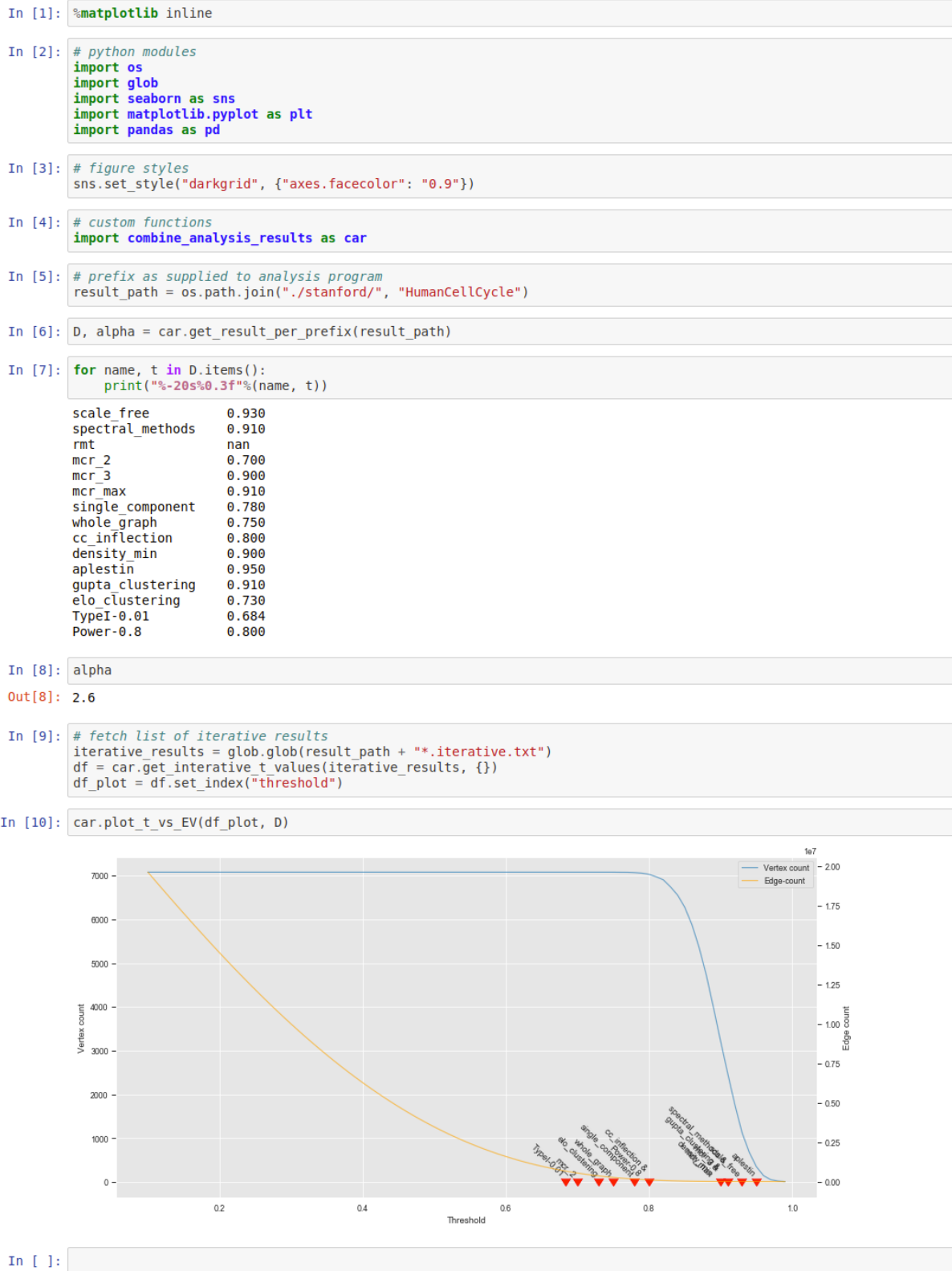


Figure 2.3: Analysis of thresholding results as viewed in an interactive Jupyter notebook.

2.3.3 Benchmarking

Example graphs were curated from online resources. In addition to the previously mentioned human cell cycle graph, a total of 181 graphs are listed in Table 2.A.1 in Appendix 2.A. The sources are as follows: EntropyExplorer – Pearson graphs of disease and control microarray data from [42], GDC – Pearson graphs of per cancer methylation β values downloaded from the Genomic Data Commons (<https://portal.gdc.cancer.gov/>) [20], IntAct - The IntAct Molecular Interaction Database networks downloaded from <https://www.ebi.ac.uk/intact/downloads> [31], ManyMicroarray – Pearson graphs of pre-normalized microarray datasets downloaded from <https://doi.org/10.5683/SP2/JOJYOP>) [25, 26].

Graphs were pre-thresholded to be smaller than 3GiB to be able to load into memory and transfer between platforms.

We applied the thresholding techniques as implemented in the software to all the graphs. In some cases, methods did not converge to a suggested threshold, or computational limits meant a method could not be completed for larger graphs (such as rmt and maximal clique based methods).

We were able to annotate vertices with gene symbols in 83 of the 181 graphs (ManyMicroarray 20, GDC 44, Entropy 19). Across these 83 graphs, 503 thresholds greater or equal to 0.60 were identified. For each of these thresholds, we applied non-overlapping paraclique to identify the 300 largest probe clusters in the graph. After 60 hours we terminated 12 paraclique calls.

The Gene Ontology (GO) project provides a systematic functional annotation of predicted gene products [40]. After converting the vertex IDs to gene symbols, we searched for enrichment of biological process GO terms in each of the paracliques with more than five genes, using the Panther web API [30]. In total, we tested 58,431 gene clusters.

To compare the thresholding methods, we scored each threshold by the number of clusters with significant GO-terms, by the average number of GO-term per cluster, and by the total number of significantly associated terms across all clusters (per graph).

For each method we quantified the effect of graph source on cluster GO-enrichment. Firstly, for each threshold method, we used a Kruskal-Wallis H-test to determine if the medians of the number of significant clusters of each of the three graph sources differed (p -value < 0.01). For those that did we used Conovers post-hoc pairwise comparison test to identify the differences [39].

For each graph we also calculated the mean, skewness, kurtosis and standard deviation of the edge weights. We used these four metrics to determine how threshold performance is related to the distribution of the edge weights using ordinary linear regression [38].

The ManyMicroarray dataset contains a ground truth graph that the authors produced by combining datasets. Based on this graph, we calculated the precision, recall, and F_1 score for each thresholded graph. The F_1 score for a graph G is given as:

$$F_1 = \left(\frac{2}{p^{-1} + r^{-1}} \right),$$

where p is the precision (number of edges in G agreeing with the ground truth graph divided by the number of edges in G) and r is the recall (number of edges in G agreeing with the ground truth graph divided by the number of edges in the ground truth graph). Using a one-sided Kolmogorov-Smirnov test, we tested if the precision, recall, or F_1 score was significantly higher than those of the alternative methods (p -value < 0.05).

2.4 Results and discussion

Firstly we consider the individual results in the human cell cycle graph, and then compared the thresholding techniques across multiple graphs to determine their performance.

Human cell cycle graph

Figure 2.4 shows in detail how a number of the thresholds are identified. Figure 2.4A shows the number of vertices per threshold. The whole-graph threshold is at the point before any vertices are removed from the graph, and cc-inflection is the point before the size of the largest connected component sharply decreases. Figure 2.4B shows selection of the smallest threshold with the maximum number of almost-disconnected components, when the Fiedler value (second eigenvalue) is “small.” Figure 2.4C shows the number of connected components per threshold. The single-component threshold is the largest threshold before multiple connected components appear. Figure 2.4D shows the density per threshold, and the minimum density. Figure 2.4E shows the clustering coefficient based methods. The elo-clustering threshold occurs when the difference

between the graphs clustering coefficient and a random counterpart graphs clustering coefficient starts to decrease. The gupta-clustering threshold occurs at the point before a sharp increase in the clustering coefficient. Figure 2.4F shows the relationship between the change in the number of vertices and the number of edges. The apeltsin threshold is the point where the number of remaining edges increases relative to the number of remaining vertices. Figure 2.4G shows the number of maximal cliques, and the points where the number of maximal cliques double and triple. Figure 2.4H shows the power curve for Pearson with significance of 0.01.

Figure 2.5 shows the relationship between vertex and edge count, density and the identified thresholds. Two grouping appear -one grouping of thresholds occur before the drop in vertices, and the other after.

The result of the spectral methods analysis of the human call cycle graph is shown in Figure 2.4B. In spectral methods, since we do not have an analytical technique to identify the size of the second eigenvalue value that ensures the existence of nearly disconnected components, we only consider the nearly disconnected component count that corresponds to a second eigenvalue of less than 1. The results of the analysis also allow us to examine the relationship between the second eigenvalue and density, as seen in Figure 2.6. This relationship has lent the name of algebraic connectivity to the second eigenvalue.

Multiple related GO-terms were found to be significantly associated with many of the clusters (False Discovery Rate < 0.05). Table 2.4 shows this result.

TypeI-0.01 error correction obtained the highest number of significant clusters, and local-global obtained the highest number of significant GO-terms. Elo-clustering and local-global methods had the same number of significant clusters, but the local-global clusters resulted in a higher number of significant GO-terms. The mean t threshold per vertex for this alpha value was 0.751, which is above the threshold to control Type I errors but not for controlling the power at 0.80 for this number of samples.

Higher thresholds resulted in a smaller graph with too few and too small clusters for significant biological meaning. These methods included scale-free, density-min, mcr-3, spectral methods, mcr-max, gupta-clustering and apeltsin. Both statistical methods resulted in significant clusters and GO-terms, and since both are directly based on the number of samples, they do not take in account the structure of the graph.

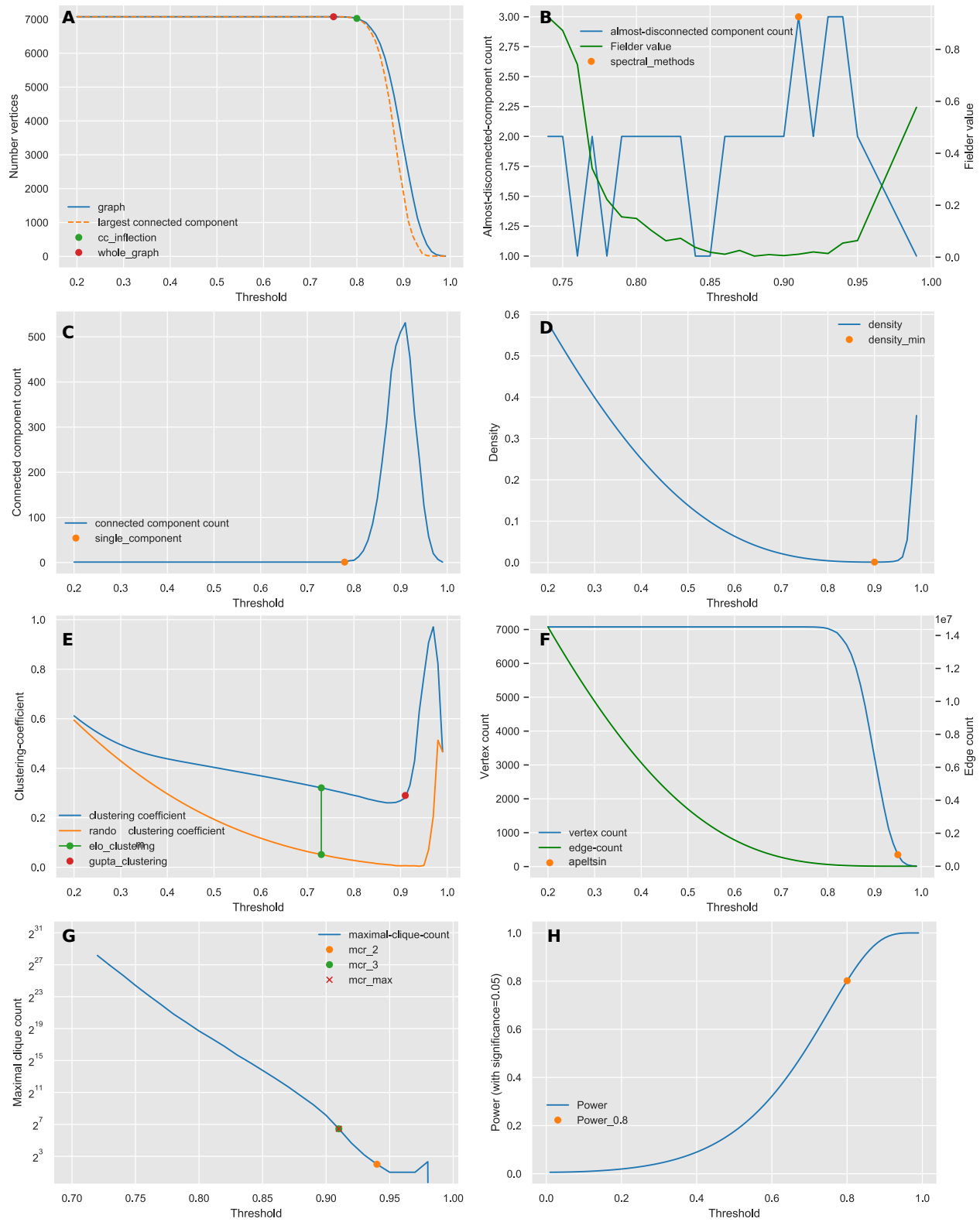


Figure 2.4: Calculation of thresholds. Detailed explanation in text.

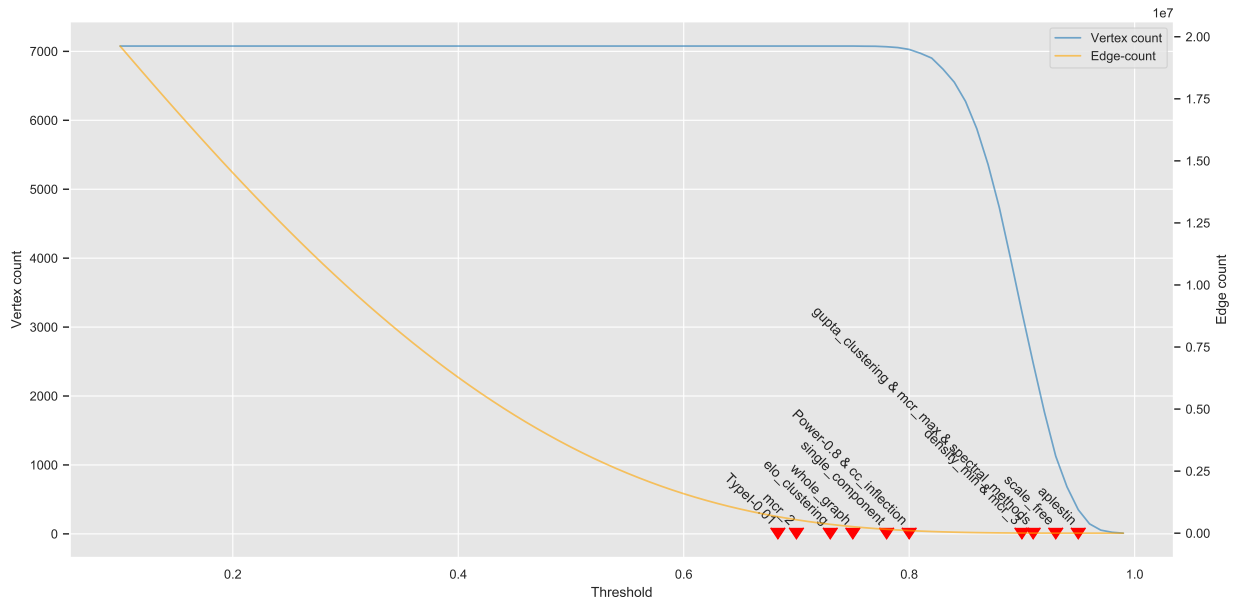


Figure 2.5: Thresholds identified for the human cell cycle graph, as they relate to both vertex and edge count.

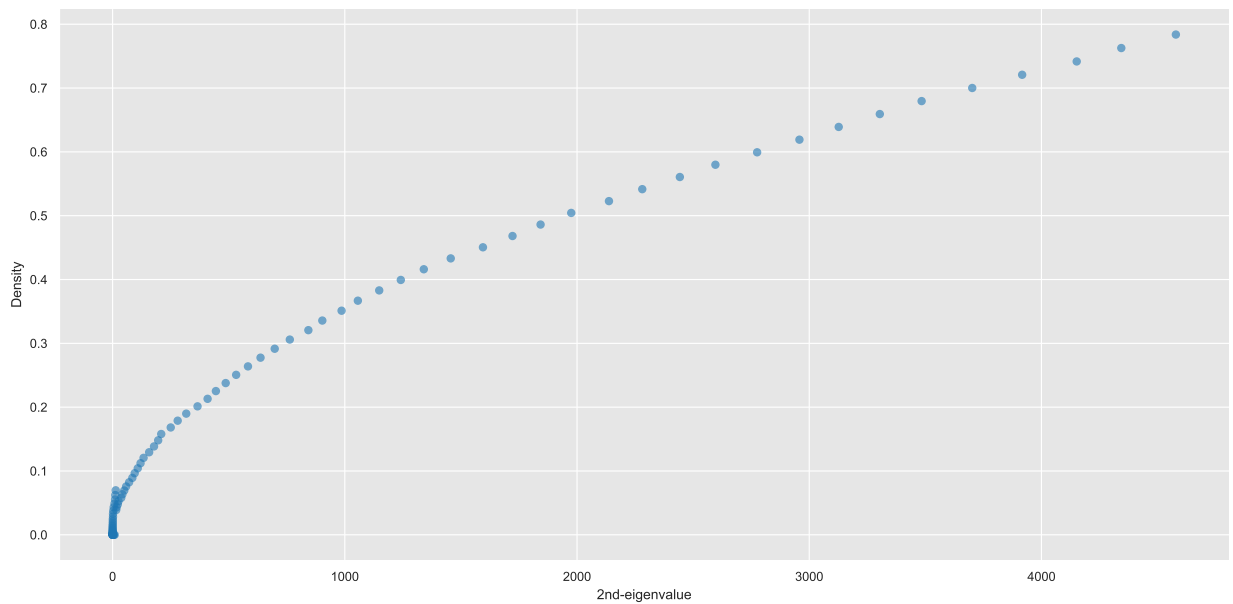


Figure 2.6: The relationship between the second eigenvalue and density in the human cell cycle graph.

Table 2.4: Total number of significant GO terms per threshold in the human cell cycle graph. Note that the threshold value in the last row refers to the average per vertex threshold corresponding to the local-global alpha of 2.60.

Threshold	Probe clusters	Gene clusters	Significant clusters	Significant GO terms	Method
0.684	300	291	26	470	TypeI-0.01
0.730	300	280	21	510	elo-clustering
0.750	300	270	18	424	whole-graph
0.780	300	242	20	253	single-component
0.800	300	224	22	256	cc-inflection
					Power-0.8
0.900	14	10	0	0	density-min
0.910	10	6	0	0	spectral-methods
					mcr-max
					mcr-3
					gupta-clustering
0.930	4	2	0	0	scale-free
0.940	3	1	0	0	mcr-2
0.950	1	0	0	0	apeltsin
0.751	300	283	21	547	local-global

Large-scale evaluations

Next, we compared identified threshold values across all the graphs within the subsets of INACT, GDC methylation, ManyMicroarray, and Entropy. The distributions of the thresholds are shown in Figure 2.7.

In the Intact graphs, no thresholds for the single-component method were identified, since the graphs started as multiple components. Additionally no thresholds could be computed with either significance method, since the number of samples were not available.

The apelsin method failed to find a threshold in the majority of the GDC and Entropy graphs, likely as a result of its assumption of skewed edge weight distribution being violated.

Note that the TypeI and Power methods only depend on the sample size, and not the graph structure - hence variation and behavior of those methods across the graphs cannot be directly compared to the other methods.

The maximal clique ratio methods identified very high thresholds - a result of the top down thresholding approach. Additionally, mcr-max is partly dependent on the lower threshold bound selection, and on large graphs such as these, this process is computationally limited.

Thresholds producing a scale free graph were unsurprisingly found to be high too, due to the effect of forcing a rigid assumption on the graph structure that does not necessarily hold.

We used the interquartile range (IQR) to compare the dispersion of the methods within graph sources. These measures give an idea of how consistent the methods are within graphs of the same biological type, source, and preprocessing workflow. We exclude the significance and power methods.

For the Intact protein interaction graphs, the lowest dispersion methods were maximal clique ratio and cc-inflection. Excluding the methods that only identified very high thresholds, for the GDC methylation graphs, density-min, followed by spectral methods and cc-inflection had the lowest dispersion. Both the ManyMicroarray and Entropy graphs are transcriptomic graphs based on micro-array data. The ManyMicroarray data represent functional connectivity of genes, while the Entropy graphs are disease related. In the ManyMicroarray graphs, density-min, spectral methods and whole-graph had the lowest dispersion, and in the Entropy graphs, density-min, spectral methods and cc-inflection had the lowest dispersion.

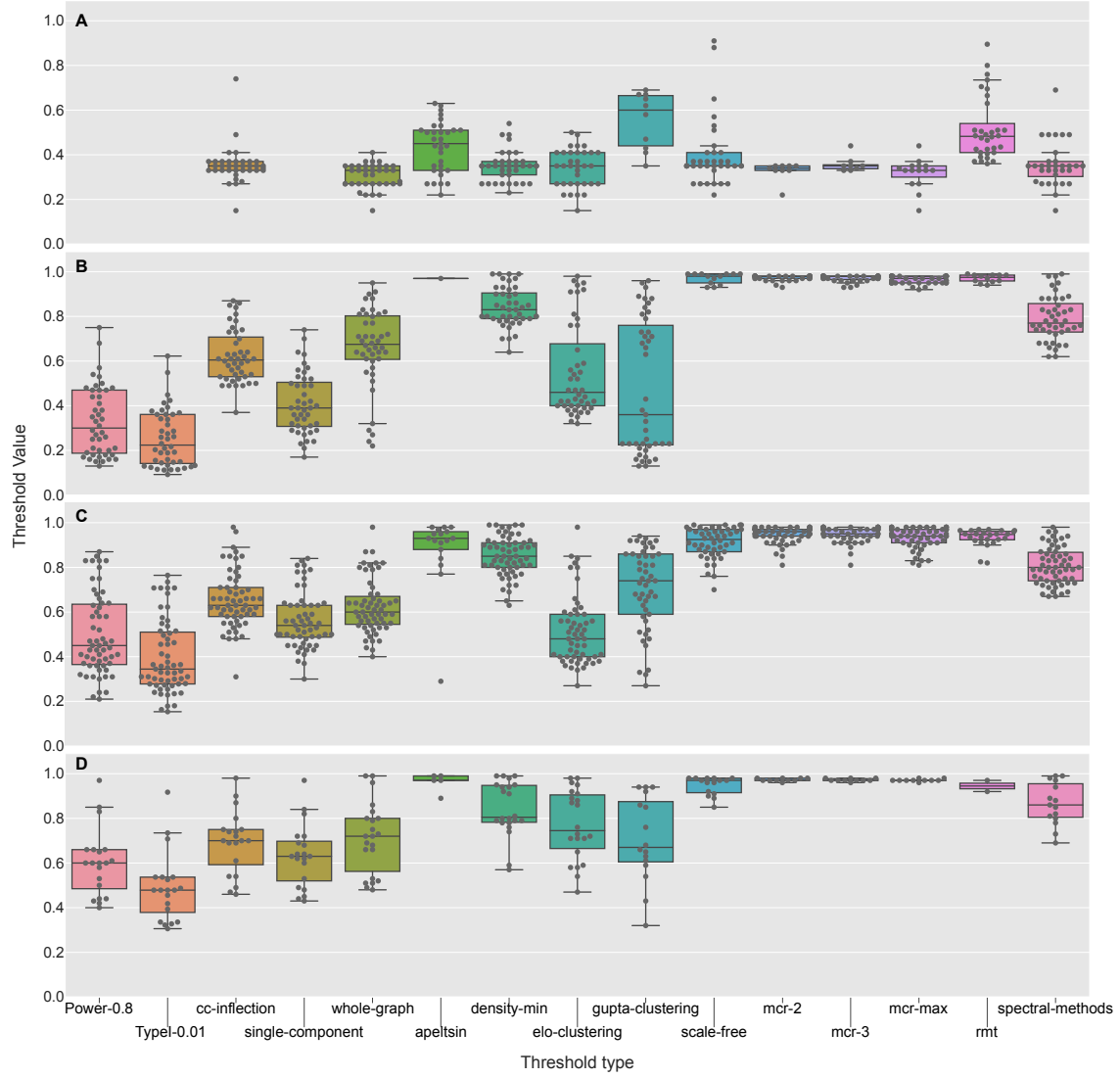


Figure 2.7: Distributions of threshold methods across the groups of graphs. (A) Intact, (B) GDC methylation, (C) ManyMicroarray, and (D) Entropy.

Table 2.5 contains the aggregate results of the cluster GO-annotation per method, aggregated by totals across all graphs and by averages across all graphs.

By totals, whole-graph resulted in the most significant clusters, cc-inflection narrowly beat whole-graph in identifying the most significant terms, and whole-graph had the highest total of average GO-terms per cluster. Spectral methods and density-min also did relatively well. By averages, single-component had the highest number of significant clusters, significant GO-terms and average significant GO-terms per cluster. cc-inflection and Power-0.8 also did above average.

Methods that did well by totals were able to identify good thresholds across all graphs, while those that did well by averages may have identified fewer thresholds, but each threshold resulted in a high quality graph. Methods that did relatively well in both cases include cc-inflection and whole-graph.

Methods with significant differences in the number of significant clusters between graph sources are cc-inflection, density-min, mcr-max scale-free, spectral-methods, and whole-graph. Boxplots of these results are shown in Figure 2.8. In each of these plots we notice that the GDC graphs have lower numbers, this may be due to GO-term enrichment being less relevant to methylation graphs in general. Disregarding the GDC graphs shows that density-min is the only method with a significant difference between the number of significant clusters identified in the Entropy and ManyMicroarray graphs.

To aid in selecting a threshold method for a particular graph, we attempted to discover how the methods perform in relation to the edge weight distribution. We used the mean (a measure of location), skewness (a measure of symmetry), kurtosis (a measure of tailedness) and standard deviation (a measure of dispersion) to describe the edge weight distributions and tested the linear regression model using in turn the number of significant clusters, the number of significant GO-terms and the average GO-terms per cluster as the dependent variable. We selected models with $R^2 > 0.35$ and $p\text{-value} < 0.01$. A select number of significant graphs are shown in Figure 2.9. These show that cc-inflection improves as skewness increases. cc-inflection, whole-graph and rmt methods decline as the edge weight mean increases. Power-0.8 also declines as kurtosis increases.

Our second approach to testing the thresholds was to compare the thresholded ManyMicroarray graphs to a ground truth graph. Figure 2.10 shows the result of comparing precision, recall and F_1 score of each method. No methods did significantly better in both recall and precision, however

Table 2.5: Results of GO-cluster analysis per threshold method.

Method	Totals across all graphs				Averages across all graphs			
	Gene clusters	Significant clusters	Significant GO-terms	Average significant GO-terms per cluster	Gene clusters	Significant clusters	Significant GO-terms	Average significant GO-terms per cluster
apelstin	19.0	14.0	795.0	548.25	1.73	1.27	72.27	49.84
cc_inflection	11,173.0	1,320.0	56,999.0	29,217.82	248.29	29.33	1,266.64	649.28
density_min	8,506.0	1,056.0	32,343.0	16,943.61	105.01	13.04	399.3	209.18
elo_clustering	2,345.0	397.0	38,059.0	19,652.86	80.86	13.69	1,312.38	677.68
gupta_clustering	5,954.0	667.0	21,480.0	11,334.36	145.22	16.27	523.9	276.45
mcr_2	789.0	100.0	2,080.0	1,337.58	12.93	1.64	34.1	21.93
mcr_3	804.0	92.0	1,721.0	1,180.33	13.63	1.56	29.17	20.01
mcr_max	1,099.0	125.0	3,063.0	1,934.5	16.4	1.87	45.72	28.87
rmt	38.0	17.0	809.0	592.75	2.71	1.21	57.79	42.34
scale_free	232.0	91.0	3,352.0	2,109.89	5.52	2.17	79.81	50.24
single_component	3,821.0	654.0	27,977.0	14,364.91	238.81	40.88	1,748.56	897.81
spectral_methods	10,367.0	1,268.0	45,908.0	23,863.7	131.23	16.05	581.11	302.07
whole_graph	13,545.0	1,546.0	56,114.0	30,708.86	250.83	28.63	1,039.15	568.68
Power-0.8	2,310.0	346.0	14,379.0	7,862.25	231.0	34.6	1,437.9	786.23

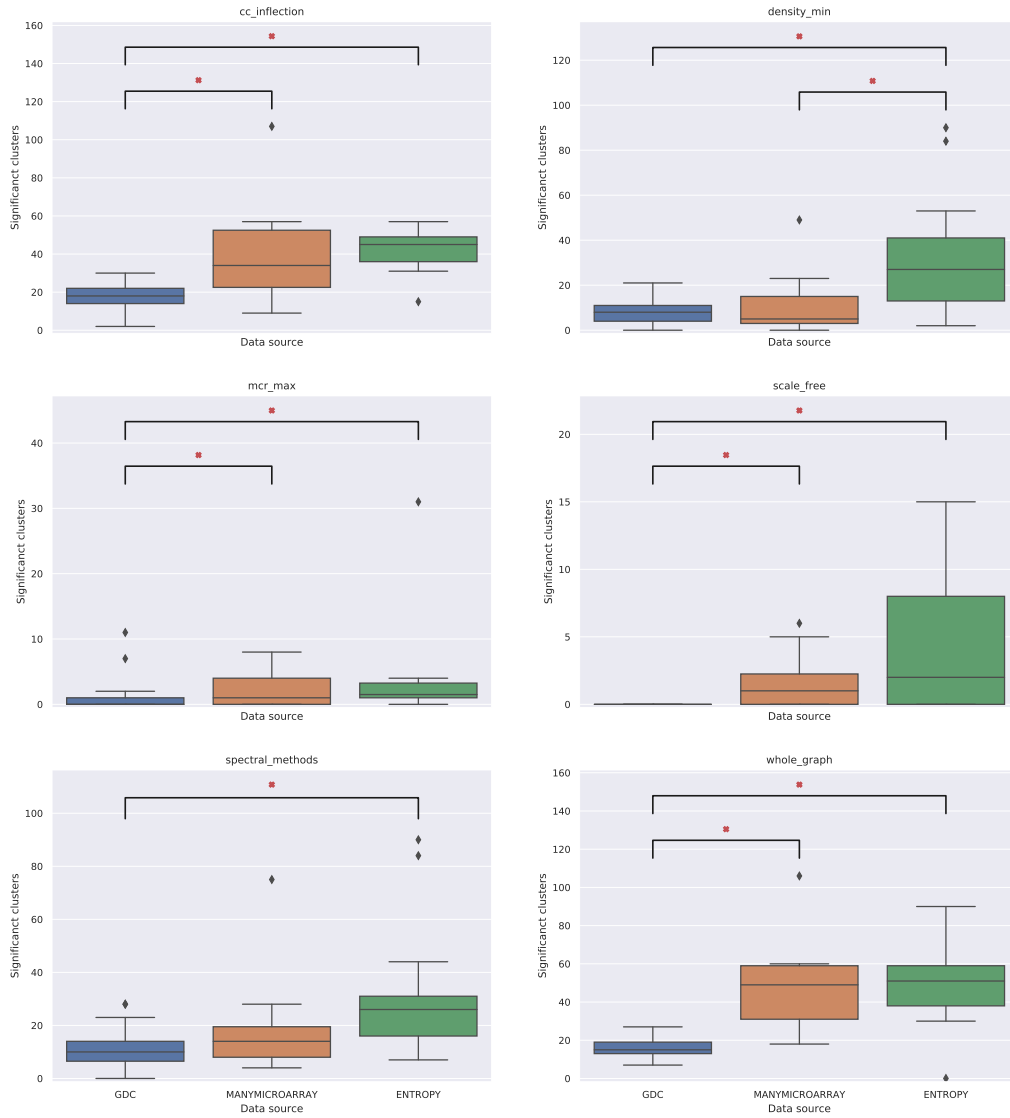


Figure 2.8: Boxplots of the number of significant clusters per data source for methods with significant differences across data sources. Lines point to data pairs with a significant difference.

a number were significantly higher recall and F_1 - implying the average was still relatively higher despite precision not being significantly high. These methods are cc-inflection, elo-clustering, single-component, whole-graph, Power-0.8, and TypeI-0.01. Methods with high precision are density-min, maximal clique bases methods, rmt, and scale-free. These methods, apart from density-min, as seen in Figure 2.7A identified the majority of thresholds above 0.90, resulting in smaller graphs, and less chance for false positives.

These are just a few approaches to testing the fidelity of the threshold to biological knowledge. Other tests, such as sensitivity to removed samples, added noise, removed and addition of random edges or vertices will also give an idea of how robust the underlying model is to noise, and therefore if the threshold holds meaning in biological applications.

2.5 Conclusion

There are a wide variety of analytical techniques that have been developed to identify a threshold value. When deciding on a threshold, it is important to consider both the biological strength, and the statistical significance of the association, and understand the underlying assumptions of the thresholding technique. For example, a scale-free based threshold may force an unnatural structure on the graph, and minimizing the density may result in a tree-like graph.

Thresholding, and further graph clustering techniques, can be used to expose structure in a dataset, and is an initial step in using network analysis and graph theory to analyze data. Once the dataset has been narrowed down to a manageable size or targeted subset, graphical models and estimation methods can be used to determine a separate graph with dependency structure, direction, and causal edges between variables.

Based on our analysis, cc-inflection fared best in producing a biologically significant graph. Therefore, unless alternative information directs a specific approach to determining a graph threshold, cc-inflection can be used as a good threshold. If consistency in graph size by way of vertex count is preferred, then the whole-graph approach also fared well across the analyses. These methods work to include the majority of vertices while still removing low valued edges, and are extremely simple to compute.

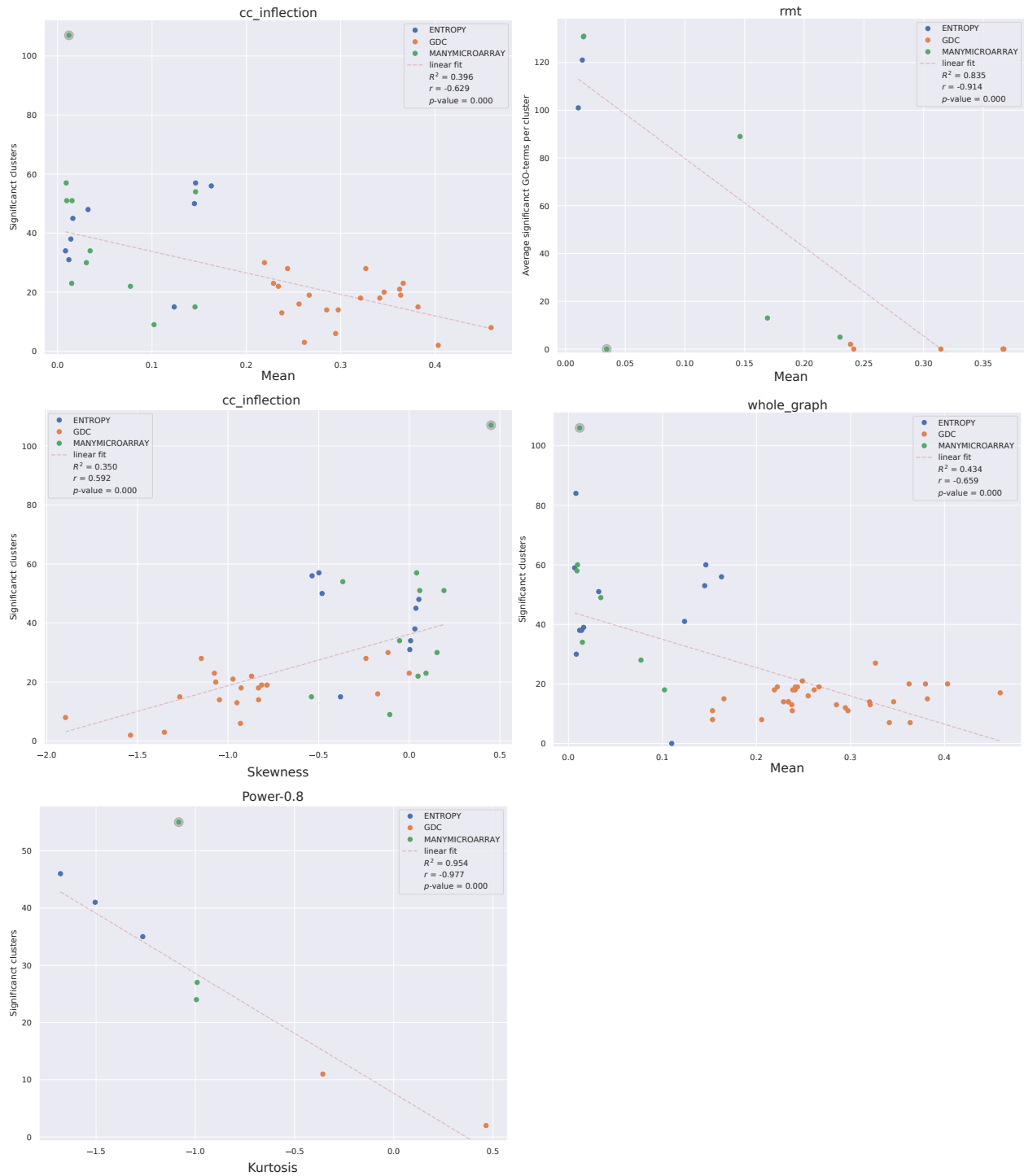


Figure 2.9: Relationships between the GO enrichment results and graph edge weight distribution measures for a select number of thresholding methods.



Figure 2.10: Precision, recall and F_1 scores of global threshold techniques in the ManyMicroarray dataset. Red asterisks denote the distribution of the score for the technique is significantly higher compared to the alternative methods.

Of course, enough samples need to be collected in order to obtain a minimum desired power or significance. When the sample size is large enough, one easily controls the statistical error rates with thresholding. Either the power or significance threshold can be used as an initial threshold, before attempting to find a threshold based on graph structure.

This research has resulted in a toolbox to allow researchers to analyze their graphs in a number of ways and produces a number of suggested thresholds. We do not make any claims on the applicability specific methods, but hope that the through descriptions, explanations of the underlying assumptions, and our recommendations based on the analyses this work will aid researchers to make an informed choice and supply them with a number of tools to easily investigate the graph behavior across many threshold values.

In conclusion, we proposed a number of novel threshold assumptions, and modified others to be more applicable to biological graphs or to be more statistically sound. We found that the scale free assumption did not fit the majority of biological graphs, and we tested the resulting thresholded graphs of the methods in a number of ways. We found that cc-inflection fared best across the tested graphs.

2.6 Acknowledgements

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231 as well as the Advanced Computing Facility (ACF) supported by the Joint Institute for Computational Sciences (JICS), the University of Tennessee Knoxville (UTK), and the University of Tennessee Health Science Center.

Bibliography

- [1] Ala, U., Piro, R. M., Grassi, E., Damasco, C., Silengo, L., Oti, M., Provero, P., and Di Cunto, F. (2008). Prediction of Human Disease Genes by Human-Mouse Conserved Coexpression Analysis. *PLoS Computational Biology*, 4(3):e1000043. [21](#)
- [2] Aoki, K., Ogata, Y., and Shibata, D. (2007). Approaches for Extracting Practical Information from Gene Co-expression Networks in Plant Biology. *Plant and Cell Physiology*, 48(3):381–390. [14](#), [22](#)
- [3] Apeltsin, L., Morris, J. H., Babbitt, P. C., and Ferrin, T. E. (2011). Improving the quality of protein similarity network clustering algorithms using the network edge weight distribution. *Bioinformatics*, 27(3):326–333. [15](#), [16](#), [22](#)
- [4] Beretta, S., Castelli, M., Gonçalves, I., Henriques, R., and Ramazzotti, D. (2018). Learning the structure of Bayesian networks: A quantitative assessment of the effect of different algorithmic schemes. *Complexity*, 2018:1–12. [20](#)
- [5] Bochkhanov, S. (2019). Alglib. <https://www.alglib.net/>. [21](#)
- [6] Borate, B. R., Chesler, E. J., Langston, M. A., Saxton, A. M., and Voy, B. H. (2009). Comparison of threshold selection methods for microarray gene co-expression matrices. *BMC research notes*, 2:240. [13](#), [19](#), [22](#)
- [7] Bornholdt, S. and Schuster, H. G. (2002). *Handbook of Graphs and Networks*. Wiley. [16](#)
- [8] Broido, A. D. and Clauset, A. (2019). Scale-free networks are rare. *Nature communications*, 10(1):1017. [11](#)
- [9] Bullmore, E. T. and Bassett, D. S. (2011). Brain Graphs: Graphical Models of the Human Brain Connectome. *Annual Review of Clinical Psychology*, 7:113–140. [14](#)
- [10] Chickering, D. M. (1996). Learning Bayesian Networks is NP-Complete. pages 121–130. Springer, New York, NY. [20](#)
- [11] Cho, R. J., Huang, M., Campbell, M. J., Dong, H., Steinmetz, L., Sapinoso, L., Hampton, G., Elledge, S. J., Davis, R. W., and Lockhart, D. J. (2001). Transcriptional regulation and function during the human cell cycle. Technical report. [23](#)

- [12] Chung, F. R. (1997). *Spectral graph theory*. Number 92. American Mathematical Soc. [11](#)
- [13] Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, New York, 2 edition. [17](#), [19](#)
- [14] Csardi, G., Nepusz, T., et al. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5):1–9. [21](#)
- [15] Derényi, I., Palla, G., and Vicsek, T. (2005). Clique Percolation in Random Networks. [13](#)
- [16] Ding, C. H. Q., He, X., and Zha, H. (2004). A spectral method to separate disconnected and nearly-disconnected Web graph components. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01*, pages 275–280, New York, New York, USA. ACM Press. [11](#)
- [17] Elo, L. L., Järvenpää, H., Orešič, M., Lahesmaa, R., and Aittokallio, T. (2007). Systematic construction of gene coexpression networks with applications to human T helper cell differentiation process. *Bioinformatics*, 23(16):2096–2103. [16](#), [17](#), [22](#)
- [18] Freeman, T. C., Goldovsky, L., Brosch, M., Van Dongen, S. M., Mazière, P., Grocock, R. J., Freilich, S., Thornton, J., and Enright, A. J. (2007). Construction, Visualisation, and Clustering of Transcription Networks from Microarray Expression Data. *PLoS Computational Biology*, 3(10):e206. [20](#)
- [19] Gibson, S. M., Ficklin, S. P., Isaacson, S., Luo, F., Feltus, F. A., and Smith, M. C. (2013). Massive-Scale Gene Co-Expression Network Construction and Robustness Testing Using Random Matrix Theory. *PLoS ONE*, 8(2):e55871. [12](#), [22](#)
- [20] Grossman, R. L., Heath, A. P., Ferretti, V., Varmus, H. E., Lowy, D. R., Kibbe, W. A., and Staudt, L. M. (2016). Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, 375(12):1109–1112. [27](#)
- [21] Gupta, A., Maranas, C. D., and Albert, R. (2006). Elucidation of directionality for co-expressed genes: Predicting intra-operon termination sites. *Bioinformatics*, 22(2):209–214. [16](#), [22](#)

- [22] Guzzi, P. H., Veltri, P., and Cannataro, M. (2014). Thresholding of Semantic Similarity Networks Using a Spectral Graph-Based Technique. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8399 LNAI, pages 201–213. Springer, Cham. [8](#), [15](#), [22](#)
- [23] Khanin, R. and Wit, E. (2006). How scale-free are biological networks. *Journal of computational biology*, 13(3):810–818. [11](#)
- [24] Langfelder, P. and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559. [11](#)
- [25] Lee, H. K., Hsu, A. K., Sajdak, J., Qin, J., and Pavlidis, P. (2004). Coexpresion analysis of human genes across many microarray data sets. *Genome Research*, 14(6):1085–1094. [18](#), [21](#), [27](#)
- [26] Lee, H. K., Hsu, A. K., Sajdak, J., Qin, J., and Pavlidis, P. (2019). Coexpression analysis of human genes across many microarray data sets. [27](#)
- [27] Luo, F., Yang, Y., Zhong, J., Gao, H., Khan, L., Thompson, D. K., and Zhou, J. (2007). Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. *BMC Bioinformatics*, 8:1–17. [12](#), [13](#), [22](#)
- [28] Luo, F., Zhong, J., Yang, Y., Scheuermann, R. H., and Zhou, J. (2006). Application of random matrix theory to biological networks. *Physics Letters A*, 357(6):420 – 423. [12](#), [22](#)
- [29] Mao, L., Van Hemert, J. L., Dash, S., and Dickerson, J. A. (2009). Arabidopsis gene co-expression network and its functional modules. *BMC Bioinformatics*, 10(1):346. [14](#)
- [30] Mi, H., Huang, X., Muruganujan, A., Tang, H., Mills, C., Kang, D., and Thomas, P. D. (2017). Panther version 11: expanded annotation data from gene ontology and reactome pathways, and data analysis tool enhancements. *Nucleic Acids Research*, 45(D1):D183–D189. [27](#)
- [31] Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N. H., Chavali, G., Chen, C., Del-Toro, N., Duesbury, M., Dumousseau, M., Galeota, E., Hinz, U., Iannuccelli, M., Jagannathan, S., Jimenez, R., Khadake, J., Lagreid, A., Licata, L.,

- Lovering, R. C., Meldal, B., Melidoni, A. N., Milagros, M., Peluso, D., Perfetto, L., Porras, P., Raghunath, A., Ricard-Blum, S., Roechert, B., Stutz, A., Tognolli, M., van Roey, K., Cesareni, G., and Hermjakob, H. (2014). The MIntAct projectIntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research*, 42(D1):D358–D363. [27](#)
- [32] Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818. [13](#)
- [33] Pellegrino, M., Provero, P., Silengo, L., and Di Cunto, F. (2004). CLOE: Identification of putative functional relationships among genes by comparison of expression profiles between two species. *BMC Bioinformatics*, 5(1):179. [21](#)
- [34] Perkins, A. D. and Langston, M. A. (2009). Threshold selection in gene co-expression networks using spectral graph theory techniques. *BMC Bioinformatics*, 10(Suppl 11):S4. [11](#), [12](#), [22](#)
- [35] Ruan, J., Dean, A. K., and Zhang, W. (2010). A general co-expression network-based approach to gene expression analysis: Comparison and applications. *BMC Systems Biology*, 4(1):8. [20](#)
- [36] Scutari, M. and Nagarajan, R. (2013). Identifying significant edges in graphical models of molecular networks. *Artificial Intelligence in Medicine*, 57(3):207 – 217. [20](#)
- [37] Scutari, M., Vitolo, C., and Tucker, A. (2018). Learning bayesian networks from big data with greedy search: Computational complexity and efficient implementation. *arXiv preprint arXiv:1804.08137*. [20](#)
- [38] Seabold, S. and Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*. [28](#)
- [39] Terpilowski, M. (2019). scikit-posthocs: Pairwise multiple comparison tests in python. *The Journal of Open Source Software*, 4(36):1169. [27](#)

- [40] The Gene Ontology, C., Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29. [27](#)
- [41] Voy, B. H., Scharff, J. A., Perkins, A. D., Saxton, A. M., Borate, B., Chesler, E. J., Branstetter, L. K., and Langston, M. A. (2006). Extracting gene networks for low-dose radiation using graph theoretical algorithms. *PLoS Computational Biology*, 2(7):0757–0768. [21](#)
- [42] Wang, K., Phillips, C. A., Saxton, A. M., and Langston, M. A. (2015). EntropyExplorer: an R package for computing and comparing differential Shannon entropy, differential coefficient of variation and differential expression. *BMC Research Notes*, page 832. [27](#)
- [43] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world’ networks. *Nature*, 393(6684):440–442. [16](#)
- [44] Zhang, B. and Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1). [10](#)
- [45] Zhou, X., Kao, M. C. J., and Wong, W. H. (2002). Transitive functional annotation by shortest-path analysis of gene expression data. *Proceedings of the National Academy of Sciences of the United States of America*, 99(20):12783–12788. [20](#)

Appendices 2

2.A Benchmark graphs

Table [2.A.1](#) contains information in the 184 graphs used in the Methods section.

Table 2.A.1: Graphs used in thresholding benchmarks.

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
ENTROPYARcase	48803	122856050	22	Pearson	0.40	ENTROPY	transcriptomic
ENTROPYARcontrol	48803	129987174	22	Pearson	0.40	ENTROPY	transcriptomic
ENTROPYAsthmaGSE4302case	54675	67633859	42	Pearson	0.50	ENTROPY	transcriptomic
ENTROPYAsthmaGSE4302control	54674	85394580	28	Pearson	0.50	ENTROPY	transcriptomic
ENTROPYBreastCancerGSE10810case	18382	77451999	31	Pearson	0.20	ENTROPY	transcriptomic
ENTROPYBreastCancerGSE10810control	18382	106804936	27	Pearson	0.20	ENTROPY	transcriptomic
ENTROPYCLLGSE8835case	22283	105753889	22	Pearson	0.20	ENTROPY	transcriptomic
ENTROPYCLLGSE8835control	22283	96058187	12	Pearson	0.30	ENTROPY	transcriptomic
ENTROPYColorectalCancerGSE9348case	54675	65357733	70	Pearson	0.30	ENTROPY	transcriptomic
ENTROPYCrohnsDiseaseGSE6731case	12625	69516918	6	Pearson	0.10	ENTROPY	transcriptomic
ENTROPYCrohnsDiseaseGSE6731control	12625	72340397	4	Pearson	0.10	ENTROPY	transcriptomic
ENTROPYLungAdenocarcinomaGSE7670case	22283	104190618	28	Pearson	0.20	ENTROPY	transcriptomic
ENTROPYLungAdenocarcinomaGSE7670control	22283	110944181	28	Pearson	0.20	ENTROPY	transcriptomic
ENTROPYPancreasGDS4102case	54613	84905193	37	Pearson	0.50	ENTROPY	transcriptomic
ENTROPYProstateCancerGSE6919GPL8300case	12625	42661568	61	Pearson	0.10	ENTROPY	transcriptomic
ENTROPYProstateCancerGSE6919GPL8300control	12625	42489524	63	Pearson	0.10	ENTROPY	transcriptomic
ENTROPYPsoriasisGSE13355case	54675	70610896	58	Pearson	0.40	ENTROPY	transcriptomic
ENTROPYPsoriasisGSE13355control	54675	62520772	58	Pearson	0.40	ENTROPY	transcriptomic
ENTROPYSchizophreniaGSE17612case	54675	43757396	28	Pearson	0.50	ENTROPY	transcriptomic
ENTROPYSchizophreniaGSE17612control	54675	75127751	23	Pearson	0.50	ENTROPY	transcriptomic

– Continued on next page

Table 2.A.1 – continued from previous page

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
ENTROPYUlcerativeColitisGSE6731case	12625	66634809	11	Pearson	0.10	ENTROPY	transcriptomic
ENTROPYUlcerativeColitisGSE6731control	12624	72478905	4	Pearson	0.10	ENTROPY	transcriptomic
GDCcancerTCGA-ACC-tumor	15907	42417005	80	Pearson	0.20	GDC	methylation
GDCcancerTCGA-BLCA-normal	15907	66172158	21	Pearson	0.30	GDC	methylation
GDCcancerTCGA-BLCA-tumor	15907	59624631	412	Pearson	0.10	GDC	methylation
GDCcancerTCGA-BRCA-normal	15905	63857033	96	Pearson	0.40	GDC	methylation
GDCcancerTCGA-BRCA-tumor	15903	41142056	782	Pearson	0.20	GDC	methylation
GDCcancerTCGA-CESC-tumor	15907	64956252	307	Pearson	0.10	GDC	methylation
GDCcancerTCGA-CHOL-tumor	15907	61952978	36	Pearson	0.20	GDC	methylation
GDCcancerTCGA-COAD-normal	15907	71573101	38	Pearson	0.20	GDC	methylation
GDCcancerTCGA-COAD-tumor	15907	68400773	296	Pearson	0.10	GDC	methylation
GDCcancerTCGA-DLBC-tumor	15907	56787082	48	Pearson	0.20	GDC	methylation
GDCcancerTCGA-ESCA-normal	15907	58315549	16	Pearson	0.30	GDC	methylation
GDCcancerTCGA-ESCA-tumor	15907	69070097	185	Pearson	0.10	GDC	methylation
GDCcancerTCGA-GBM-tumor	15907	72794777	140	Pearson	0.10	GDC	methylation
GDCcancerTCGA-HNSC-normal	15907	64120869	50	Pearson	0.20	GDC	methylation
GDCcancerTCGA-HNSC-tumor	15907	65550344	528	Pearson	0.10	GDC	methylation
GDCcancerTCGA-KICH-tumor	15907	69964563	66	Pearson	0.10	GDC	methylation
GDCcancerTCGA-KIRC-normal	15907	58867721	160	Pearson	0.20	GDC	methylation
GDCcancerTCGA-KIRC-tumor	15907	65762648	319	Pearson	0.10	GDC	methylation
GDCcancerTCGA-KIRP-normal	15907	54244194	45	Pearson	0.20	GDC	methylation

– Continued on next page

Table 2.A.1 – continued from previous page

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
GDCcancerTCGA-KIRP-tumor	15906	44300164	275	Pearson	0.20	GDC	methylation
GDCcancerTCGA-LGG-tumor	15907	68191694	516	Pearson	0.10	GDC	methylation
GDCcancerTCGA-LIHC-normal	15907	63651756	50	Pearson	0.20	GDC	methylation
GDCcancerTCGA-LIHC-tumor	15907	65594030	377	Pearson	0.10	GDC	methylation
GDCcancerTCGA-LUAD-normal	15907	70180790	32	Pearson	0.30	GDC	methylation
GDCcancerTCGA-LUAD-tumor	15907	48257557	458	Pearson	0.20	GDC	methylation
GDCcancerTCGA-LUSC-normal	15907	55830090	42	Pearson	0.20	GDC	methylation
GDCcancerTCGA-LUSC-tumor	15907	60020010	370	Pearson	0.10	GDC	methylation
GDCcancerTCGA-MESO-tumor	15907	70310302	87	Pearson	0.10	GDC	methylation
GDCcancerTCGA-PAAD-tumor	15907	41444911	184	Pearson	0.20	GDC	methylation
GDCcancerTCGA-PCPG-tumor	15907	49263478	179	Pearson	0.20	GDC	methylation
GDCcancerTCGA-PRAD-normal	15907	56154376	50	Pearson	0.20	GDC	methylation
GDCcancerTCGA-PRAD-tumor	15907	70950680	498	Pearson	0.10	GDC	methylation
GDCcancerTCGA-READ-tumor	15907	45016539	98	Pearson	0.20	GDC	methylation
GDCcancerTCGA-SARC-tumor	15907	64223653	261	Pearson	0.10	GDC	methylation
GDCcancerTCGA-SKCM-tumor	15907	42028014	104	Pearson	0.20	GDC	methylation
GDCcancerTCGA-STAD-tumor	15907	48578537	395	Pearson	0.20	GDC	methylation
GDCcancerTCGA-TGCT-tumor	15907	59385135	150	Pearson	0.20	GDC	methylation
GDCcancerTCGA-THCA-normal	15907	68540746	56	Pearson	0.20	GDC	methylation
GDCcancerTCGA-THCA-tumor	15907	64070850	507	Pearson	0.10	GDC	methylation
GDCcancerTCGA-THYM-tumor	15907	71023797	124	Pearson	0.10	GDC	methylation

– Continued on next page

Table 2.A.1 – continued from previous page

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
GDCcancerTCGA-UCEC-normal	15905	55327327	46	Pearson	0.30	GDC	methylation
GDCcancerTCGA-UCEC-tumor	15907	40929200	431	Pearson	0.20	GDC	methylation
GDCcancerTCGA-UCS-tumor	15899	38584220	57	Pearson	0.20	GDC	methylation
GDCcancerTCGA-UVM-tumor	15907	67197278	80	Pearson	0.10	GDC	methylation
HumanCellCycle	7077	25038426	13	Pearson	0.10	Stanford	transcriptomic
INTACTarath	8199	38235	- -		-	INTACT	proteomic
INTACTbacsu	683	1546	- -		-	INTACT	proteomic
INTACTbovin	374	462	- -		-	INTACT	proteomic
INTACTcaeel	12393	38922	- -		-	INTACT	proteomic
INTACTcamje	1311	11556	- -		-	INTACT	proteomic
INTACTchatd	67	104	- -		-	INTACT	proteomic
INTACTchemic	254	225	- -		-	INTACT	proteomic
INTACTchick	247	239	- -		-	INTACT	proteomic
INTACTchlre	55	77	- -		-	INTACT	proteomic
INTACTdanre	346	458	- -		-	INTACT	proteomic
INTACTdesvh	679	853	- -		-	INTACT	proteomic
INTACTdicdi	152	161	- -		-	INTACT	proteomic
INTACTdrome	11665	43211	- -		-	INTACT	proteomic
INTACTebvg	49	64	- -		-	INTACT	proteomic
INTACTecoli	3274	17801	- -		-	INTACT	proteomic
INTACTecolx	76	81	- -		-	INTACT	proteomic

– Continued on next page

Table 2.A.1 – continued from previous page

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
INTACTgerhy	14	97	-	-	-	INTACT	proteomic
INTACThelpy	982	2329	-	-	-	INTACT	proteomic
INTACThhv1	48	114	-	-	-	INTACT	proteomic
INTACThhv3	29	65	-	-	-	INTACT	proteomic
INTACThhv8	26	65	-	-	-	INTACT	proteomic
INTACThuman	28742	270235	-	-	-	INTACT	proteomic
INTACThumhe-	34	61	-	-	-	INTACT	proteomic
INTACTlambd	47	95	-	-	-	INTACT	proteomic
INTACTmetth	70	59	-	-	-	INTACT	proteomic
INTACTmouse	9607	25186	-	-	-	INTACT	proteomic
INTACTmuhv1	71	182	-	-	-	INTACT	proteomic
INTACTmycoba	70	136	-	-	-	INTACT	proteomic
INTACTmycpn	419	847	-	-	-	INTACT	proteomic
INTACTmyctu	119	122	-	-	-	INTACT	proteomic
INTACTmyxxd	87	432	-	-	-	INTACT	proteomic
INTACTorysj	219	237	-	-	-	INTACT	proteomic
INTACTpea	50	58	-	-	-	INTACT	proteomic
INTACTpethy	23	78	-	-	-	INTACT	proteomic
INTACTplaf7	1311	2777	-	-	-	INTACT	proteomic
INTACTpseae	81	70	-	-	-	INTACT	proteomic
INTACTpyrfu	87	99	-	-	-	INTACT	proteomic

– Continued on next page

Table 2.A.1 – continued from previous page

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
INTACTpyrho	115	105	-	-	-	INTACT	proteomic
INTACTrabit	100	142	-	-	-	INTACT	proteomic
INTACTrat	2648	3853	-	-	-	INTACT	proteomic
INTACTrhopa	58	57	-	-	-	INTACT	proteomic
INTACTricsi	160	284	-	-	-	INTACT	proteomic
INTACTsalty	77	85	-	-	-	INTACT	proteomic
INTACTschpo	1007	1854	-	-	-	INTACT	proteomic
INTACTstrpn	189	233	-	-	-	INTACT	proteomic
INTACTsyny3	1915	3367	-	-	-	INTACT	proteomic
INTACTtheeb	56	59	-	-	-	INTACT	proteomic
INTACTtheko	129	128	-	-	-	INTACT	proteomic
INTACTthet2	62	59	-	-	-	INTACT	proteomic
INTACTthet8	62	62	-	-	-	INTACT	proteomic
INTACTtrep	713	3553	-	-	-	INTACT	proteomic
INTACTtryb2	64	142	-	-	-	INTACT	proteomic
INTACTvaccw	49	51	-	-	-	INTACT	proteomic
INTACTvzvo	69	322	-	-	-	INTACT	proteomic
INTACTxenla	615	801	-	-	-	INTACT	proteomic
INTACTyeast	6316	82256	-	-	-	INTACT	proteomic
MANYMICROARRAYallander-gist	1987	1505893	19	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYarmstrong-mll	8546	24810832	72	Pearson	0.10	MANYMICROARRAY	transcriptomic

– Continued on next page

Table 2.A.1 – continued from previous page

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
MANYMICROARRAYbhattacharjee-lung	8753	26968955	203	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYbittner-melanoma	9297	21860091	38	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYbleharsky-leprosy	8527	29287873	11	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYbutte-relevance	5008	7542168	68	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYchang-fibroblasts	31791	156270100	50	Pearson	0.30	MANYMICROARRAY	transcriptomic
MANYMICROARRAYchaussabel-parasite	8508	26453874	28	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYchen-liver	15915	74461300	207	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYcheok-leukemia	8819	26249897	120	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYcoller-telomerase	3581	5058558	20	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYdhanasekaran-prstate	8733	28201445	53	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYdiehn-cd28	32627	148545520	68	Pearson	0.30	MANYMICROARRAY	transcriptomic
MANYMICROARRAYdyrskjot-bladderrecur	4774	8236939	31	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYdyrskjot-bladderstage	4865	8114158	40	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYgarber-lung	16377	85251409	73	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYgolub-leukemia	4822	7387644	72	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYgreenberg-myopathy	8819	30975714	12	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYgruvberger-estrogen	3369	4649407	58	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYhedenfalk-breast	3226	3803419	22	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYhedenfalk-familial	4754	8451702	16	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYhoumard-obesity	15597	81682432	24	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYhuang-breast	8788	20233477	89	Pearson	0.10	MANYMICROARRAY	transcriptomic

– Continued on next page

Table 2.A.1 – continued from previous page

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
MANYMICROARRAYhuang-papillary	8426	26997521	16	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYjazaeri-breast	6445	14124666	61	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYkhan-bluecell	2276	1708707	88	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYkhatua-astrocytoma	8396	27911970	13	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYleung-gastric	37066	91859293	126	Pearson	0.30	MANYMICROARRAY	transcriptomic
MANYMICROARRAYluo-prostate	6500	15240833	25	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYma-breast	1940	1213200	61	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYmacdonald-mblastoma	1377	678681	31	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYnielsen-sarcoma	5520	10257438	46	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYperou-breast	5519	11926719	26	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYperou-portraits	8782	23559594	84	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYpomeroy-embryonal	4931	8212744	90	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYramaswamy-cancer	11187	45525936	280	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYrickman-glioma	4921	8198480	51	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYrosenwald-leukemia	12284	52933525	102	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYross-breast	15471	95209420	24	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYross-nci60	6630	14022808	68	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYshipp-dlbcl	4842	8518902	77	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYsibille-pfc	15597	90123848	75	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYsingh-prostate	8557	30291539	102	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYsorlie-breast	15326	83653688	85	Pearson	0.10	MANYMICROARRAY	transcriptomic

– Continued on next page

Table 2.A.1 – continued from previous page

Graph name	$ V $	$ E $	n	Correlation	pre- t	Source	Datatype
MANYMICROARRAYsorlie-breasttumor	33402	151538345	122	Pearson	0.30	MANYMICROARRAY	transcriptomic
MANYMICROARRAYspannhake-asthma	8836	31701721	12	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYstaunton-nci60	4918	6747999	60	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYsu-human	8727	23991368	85	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYtezak-jdm	4586	8934426	10	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYvahey-hiv	4819	8678293	30	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYvantveer-breast	16783	77444142	117	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYvirtaneva-aml8	4037	5849781	21	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYwelle-muscle	8771	30121294	12	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYwelsh-ovarian	4589	7146487	49	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYwelsh-prostate	8602	27227860	55	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYwest-breast	4922	8438291	49	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYwhitfield-cellcycle	33282	142785555	114	Pearson	0.30	MANYMICROARRAY	transcriptomic
MANYMICROARRAYyeoh-leukemia	8823	14475994	248	Pearson	0.10	MANYMICROARRAY	transcriptomic
MANYMICROARRAYyoon-p53	4989	11964059	12	Pearson	0.10	MANYMICROARRAY	transcriptomic

Chapter 3

Magnetoencephalographic Data Analysis: New Tools for Interpreting Neuromagnetic Profiles of Resting Brain Networks in Pre-adolescent Children

Abstract

Abstract: Neural activity in the brain produces a flow of current, and therefore an electrical field around the head. Magnetoencephalography (MEG) technology uses an array of sensors arranged in a helmet placed over the head that measure minuscule changes in the magnetic fields, and therefore measures neural activity in the brain. We computed 40 brain graphs for 20 individuals each from resting state MEG measurements. Using these graphs, we were able to identify differences between individuals, brainwaves, and short- and long-term functional connectivity.

3.1 Introduction

Functional brain data

Functional brain data is a measurement of activity within the brain. Neural activity in the brain produces a flow of current and therefore electrical activity (dipoles). While we cannot, in general, directly measure the current, we can measure its by-products to obtain a glimpse of brain activity. At the same time, complementary structural measurements may be taken.

Functional brain data collection technologies include functional magnetic resonance imaging (fMRI) where the person is placed in a strong magnetic field which excites hydrogen atoms in the tissue and creates radio frequency signal, electroencephalogram (EEG) where electrodes placed on scalp measure voltage fluctuations between pairs of electrodes placed on the scalp, and magnetoencephalography (MEG), where sensors arranged in a helmet placed over the head measure magnetic fields around the brain.

MEG uses an array of sensors (usually about 300). Figure [3.1](#) illustrates the MEG technology. Maxwell's equations imply an electrical current produces an orthogonally oriented magnetic field. These sensors can measure minuscule changes in the magnetic fields, and therefore directly measures the neural activity in the brain. Approximately 50,000 active neurons (synchronized and reinforcing each other) are needed for a detectable signal.

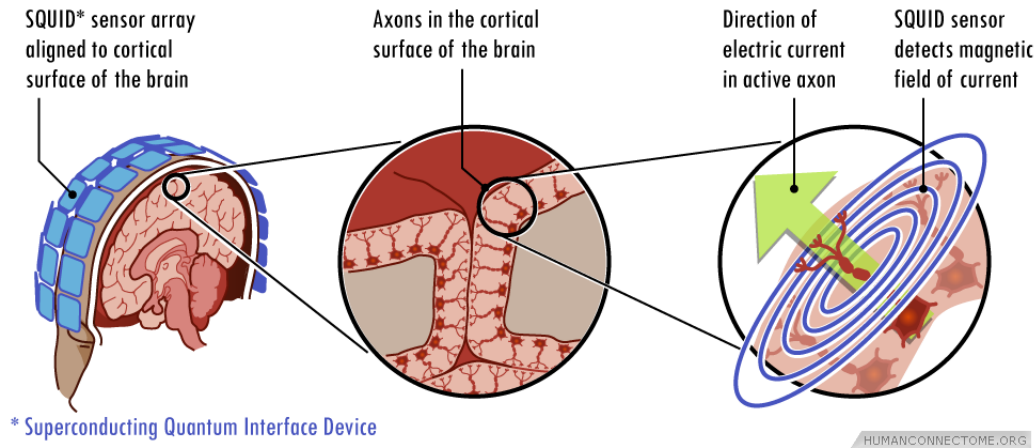


Figure 3.1: MEG sensors arranged in a helmet around the head are able to measure minute changes in the magnetic field around the brain. Figure from [7].

Resting state brain networks

Over the course of the past decade, there has been an increased effort towards understanding brain functional architecture, specifically in relation to resting brain networks (e.g. [11]). Most studies of this type involve functional magnetic resonance recording fMRI, some of which aim to establish connectivity patterns during rest, others during performance of a task, and few, under both conditions. Most readily interpretable are the former studies in that they reflect only the intrinsic state of the brain, as opposed to that of task-specific activation which is affected by a vast number of factors that define the task (such as complexity and number of cognitive operations it involves), and the various ways particular individuals approach it, many of which may be intractable. Resting state connectivity networks are of two kinds: resting functional networks, potentially, but not necessarily, corresponding to brain systems subserving classes of sensory, motor, and cognitive functions that are quantifiable using both neurophysiological (MEG/EEG) and hemodynamic (PET/fMRI) imaging techniques; and the default mode network, mainly derived using fMRI, consisting of brain areas the activity of which is suppressed during deliberate performance of specific tasks (e.g., [11]), and which may reflect the stream of resting consciousness or the performance of automated unconscious tasks ([2, 4, 22]).

The functional relevance of these networks are interpreted, primarily, based on their spatial profile and include: default mode; medial, lateral, and parietal visual; sensorimotor; temporal; executive function; cerebellum; auditory; and frontoparietal networks. However, as is inherent to hemodynamic imaging modalities, the rich spatial profile attainable with fMRI is limited in the temporal information which it can yield. One approach for obtaining complementary information regarding neural network dynamics is the use of direct non-invasive neurophysiological techniques, such as MEG. Given the millisecond resolution of the latter, it may be possible to utilize MEG to characterize, in more detail, brain networks by deriving separate connectivity maps for short-range, mid-range, and long-range connections, as typical resting brain connectivity patterns follow the so-called small-world topology characterized by a dense network of local connections between neighboring nodes and relatively few long-range connections between any (distant) pair of nodes (e.g., [19, 20]). As well as providing insight into the intrinsic state of the brain while at rest, deviations from these patterns of activity may have the potential to distinguish individuals with various disorders in the absence of gross anatomical abnormalities, such as schizophrenia [25]; major depression [3]; autism spectrum disorder [15], and Post-Traumatic Stress Disorder [18]

Aim

A fairly recent fMRI study published by Thornburgh et al. [21], reported on salient resting brain networks mapped reliably in a small and diverse cohort of children recruited from the Conditions Affecting Neurocognitive Development and Learning in Early Childhood (CANDLE) study (www.candlestudy.com) cohort enrolled between 2007 and 2011. In the present study, we seek to obtain complementary information regarding the architecture of brain networks in the same cohort, utilizing the MEG recordings rich in temporal information. Specifically, the purpose of this study is to establish sensor-space normative profiles of resting brain networks, in a cohort of homogeneous, 6- to 7-year old children, using MEG in combination with graph theoretical tools. Given the emerging utility of this methodology as a diagnostic adjunct in adults, identifying the normative profile (and normal range of variation) of brain activity during rest in pre-adolescents may help elucidate the degree to which aberrations in these networks distinguish individuals with neurodevelopmental disorders from normally developing children, early in childhood.

3.2 Methods

Study participants

The participants were 20 children who participated in the CANDLE study. The children had an average age of 6.10 years at the time of the study; 47.2% were African American, and 50.0% were boys. Participants were normally developing children with no neurological conditions, and all were right-handed.

Data acquisition procedure

All MEG recordings were conducted using a whole-head neuromagnetometer containing an array of 248 sensors (MAGNES 3600 WH, 4D Neuroimaging). Readings were taken in the sound-damped and magnetically-shielded Magnetic Source Imaging Laboratory at Le Bonheur Childrens Hospital. Participants were required to stay at rest during the recordings, without being exposed to external stimulation of any kind. Ten minutes of artifact-free continuous MEG resting data were collected for each participant during a 45 minute recording session (which included subject preparation). Head position was monitored before and after the data acquisition using a set of five sensors.

Data preparation

The physiological artifacts and other types of noise such as the electro-cardiogram, eye movements, power supply bursting and $1/f$ - like environmental noise [14] were identified using independent components analysis (ICA) and then removed from the MEG data [10], using Brainstorm (<https://neuroimage.usc.edu/brainstorm/>), a collaborative, open-source set of software tools running in Matlab. The first second of each recording was removed in an effort to ensure steady state. The floating baseline was removed by calculating the moving average, smoothing it by estimating a polynomial fit, and removing this from the original recording. Bad channels were removed using nearest neighbor similarity: the Welch power spectral density [24] of each channel was compared to its neighbors using a sum of squared errors, and those with errors three standard deviations above the mean were marked as “bad” and removed from further analysis across all

subjects to keep resultant graphs consistent across all participants. This resulted in the removal of 19 sensors from the subsequent analysis. To reduce the effects of nearby power lines, a band stop filter at 60, 120, 180 and 240 Hz was applied. Finally, band pass filters as in Table 3.1 were applied to obtain separate time series for each frequency band [8].

Graph creation

Weighted Lag Phase Index (WPLI) [23] is based on phase lag index (PLI):

$$\psi := |\mathbb{E}[\text{sign}(\mathcal{I}(X))]| \leq 1,$$

the expected value of the sign of the imaginary component of the cross-spectrum of two sensors.

WPLI is defined as:

$$\phi := \frac{|\mathbb{E}[\mathcal{I}(X)]|}{\mathbb{E}[|\mathcal{I}(X)|]}.$$

The expected values are calculated by averaging epochs. Both PLI and WPLI are measurements of the degree that phase lags and leads between two sensors are not equally probable (nonequiprobable). WPLI was developed to increase sensitivity to phase-synchronization, and be less sensitive to volume-conduction (the effect on the sensor of the conduction of electricity through the participant's head).

This index was calculated between each pair of sensors, within each bandpass for a number of epoch lengths (60, 30, 10, 5, and 2 seconds). Graphs were created using sensors as vertices, and wPLI as edge weights between vertices. This resulted in 40 graphs per participant, and a total of 800 graphs. Figure 3.2 shows the distributions of wPLI per epoch length. Next all edges between neighboring sensor were removed (resulting in a graph of 214 vertices and 22,350 edges), and then thresholding (removing low values edges) was considered. Previous thresholds that have been used includes arbitrary values [16], or based on permutation null distributions [9]. A low permutation of 1,000 would require an additional 36,465,600,000 wPLI calculations, an infeasible number for general research, and therefore we determined to come up with a different method.

Ensuring all sensors remain in each graph allows direct comparison between the graphs without the need for normalization of any metrics or consideration of missing vertices. The threshold we

Table 3.1: Band pass filters

Name	Bandpass (Hz)
δ	0.5 - 4
θ	4 - 8
α_1	8 - 10
α_2	10 - 13
β_1	13 - 15
β_2	15 - 19
β_3	20 - 29
γ	30 - 45

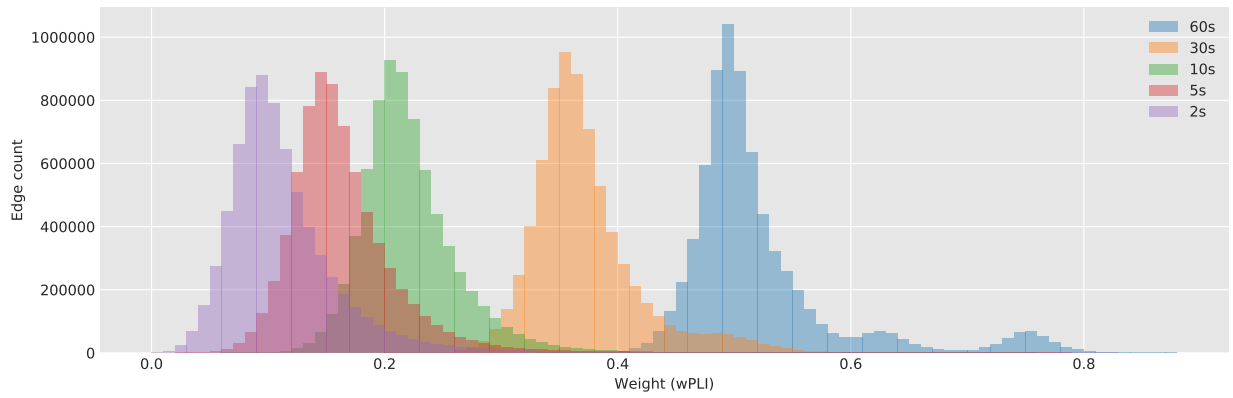


Figure 3.2: WPLI distributions for different epoch lengths.

consider therefore is the lowest value that retains all sensors in the graph. This is similar to the method used in [5], except we allow multiple connected components instead of enforcing a single connected component. Figure 3.3 shows three examples of the weight distribution and threshold calculation. Each horizontal line segment represents the range of weights of a single vertex's adjacent edges. The vertical line represents the threshold for the graph - a threshold that ensures each vertex retains at least one edge.

We classified sensors according to the brain region they correspond to on the scalp. Figure 3.4 is a key to the regions, similar to those described in [13].

Graph metrics

Previous studies have used a handful of simple graph topological metrics to compare patients. Here we are interested in such graph metrics, but also in more sophisticated graph theoretical methods. As in previous studies, we calculated the following graph metrics:

1. Density: $d = \frac{|E|}{\frac{1}{2}|V|(|V| - 1)}$.
2. Diameter: the longest shortest path of the graph.
3. Average degree centrality: $c_d = \frac{1}{|V|} \sum_i k_i$, where k_i is the degree of vertex i .
4. Average vertex betweenness centrality: $c_b = \frac{1}{|V|} \sum_i b_i$, where $b_i = \sum_{j,h(j \neq i \neq h)} \frac{P_{j,h}(i)}{P_{j,h}}$, and $P_{j,h}$ is the number of paths between vertices j and h and $P_{j,h}(i)$ is the number of paths between vertices j and h that passes through vertex i .
5. Average edge betweenness centrality: the average number of times a shortest path passes through an edge.
6. Average global efficiency: $e_g = \frac{1}{|V|} \sum_i g_i$ where $g_i = \frac{\sum_{j(j \neq i)} d_{i,j}^{-1}}{|V| - 1}$ and $d_{i,j}$ is the distance between vertex i and vertex j .
7. Average local efficiency: $e_l = \frac{1}{|V|} \sum_i l_i$ where $l_i = \frac{\sum_{j,h(j,h \in G_i, j \neq i)} d_{j,h}^{-1}}{k_i(k_i - 1)}$, and G_i is the subgraph induced by vertex i and its neighbors.

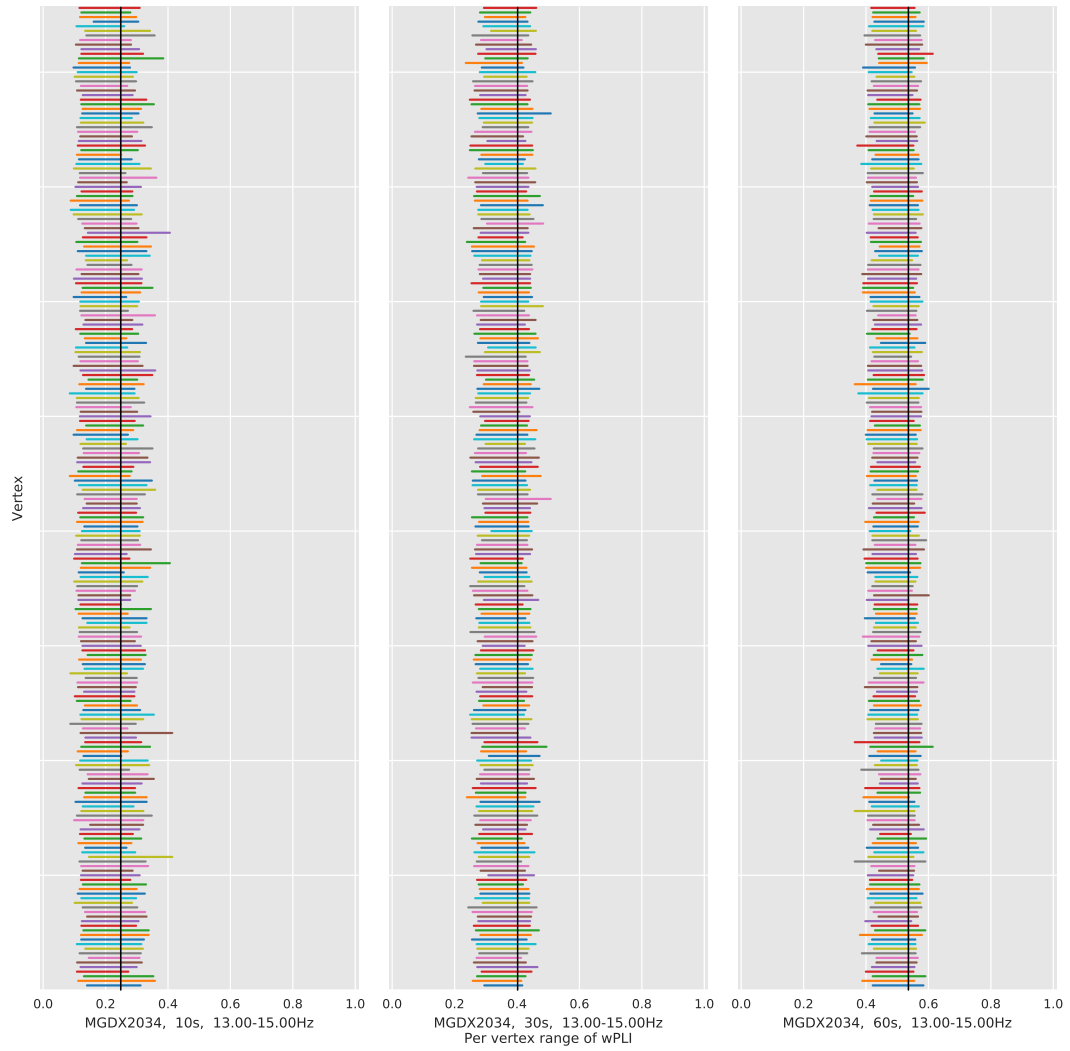


Figure 3.3: wPLI distributions for different epoch lengths. Each horizontal line represents a sensor, and the beginning and end ranges of the weights of its adjacent edges.

Cluster comparison

A clique is a subgraph that contains each possible edge between its vertices. A maximum clique is the largest possible such clique in the graph. Paraclique is a clique-centric clustering method used to reduce the effects of noise in real-world data [6]. Paraclique begins with a maximum clique, then adds highly connected vertices in a tightly controlled manner [12]. For example, a vertex is only added to the cluster as long as it is adjacent to at least all but one of the vertices in the initial maximum clique. Paracliques help identify highly correlated clusters of sensors without the need for prior knowledge.

Graph comparison

We employed the Jaccard similarity index for sets to define an analogous similarity index for graphs. Thus, given a pair of graphs we divide the number of shared edges by the total number of edges in both graphs. The result of this is a secondary graph of subjects as vertices, with edges based on their similarity of their MEG graphs. This graph can be clustered, and the clusters tested for enrichment in screening variables. If such enriched clusters exist, they are evidence of differential functional brain connectivity.

3.3 Results and discussion

The majority of graphs had a diameter of four or five (726 graphs out of the total of 800). Two graphs had the highest diameter of seven. The diameter represents the shortest path between the two most distant vertices. A low value indicates the graph is well connected, and no vertices are particularly isolated.

Average local efficiency is shown in Figure 3.5. The α_1 bandpass has the highest mean for each epoch length, and γ the lowest. The opposite is true for average edge betweenness in Figure 3.6. Local efficiency is a measure of information dissemination in the network at the local level. Edge betweenness is a measure of how often an edge has to be traversed when traveling between any two vertices in the graph. High local efficiency means vertices have well connected neighbors,

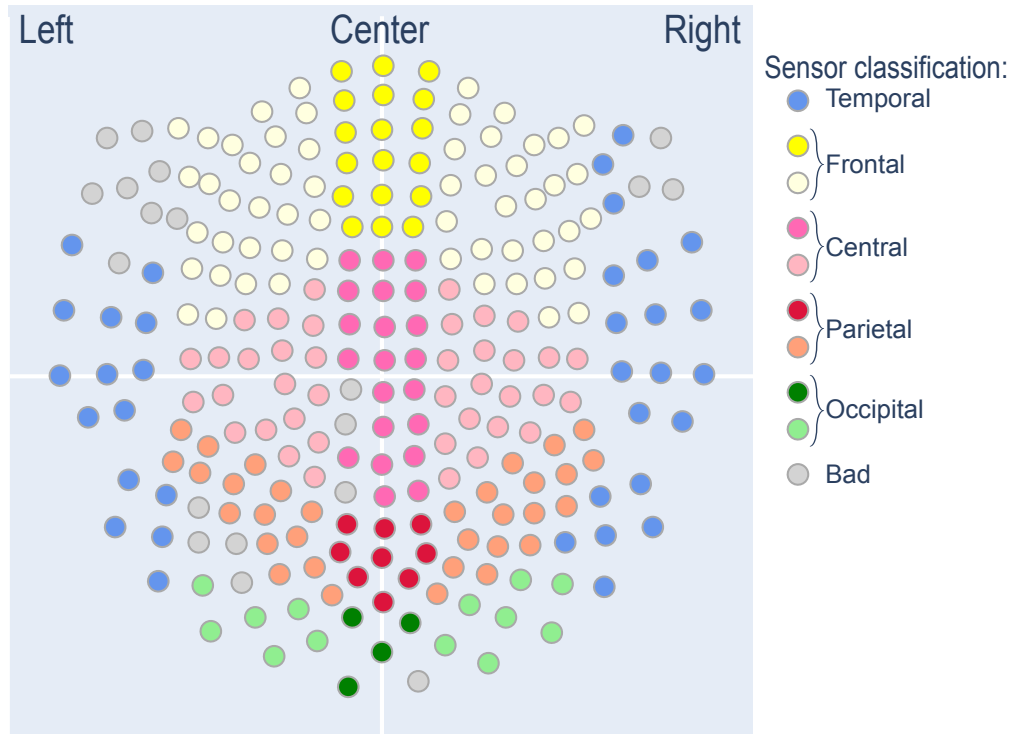


Figure 3.4: Vertex (sensor) classification according to scalp position.

high edge betweenness means many bridge (aka cut) edges exist that connect different parts of the network.

We used Euclidean distance (transformed to similarity) on the standardized graph metrics to determine how participants relate to one another in graph metric space. Figure 3.7 shows the similarity between participants based on graph metrics. As the threshold is increased we see outliers emerging, in particular MGD_X2010 and MGD_X2015. In general however, the participants are highly similar.

The average paraclique size was 7.98 vertices, with a maximum size of 29. No paracliques were completely shared by any graphs. On average, the highest cluster agreement (per bandpass and epoch length) was less than 50%. Figure 3.8 shows per vertex paraclique membership overlap across individuals. High overlap indicates the vertex occurred in a paraclique in the majority of individuals. Zero overlap indicates the vertex did not occur in a paraclique in any individual. Vertices with high overlap are indicative that they are consistently important in the functional connectivity clusters. As mentioned, we did not find clusters consistent across individuals,

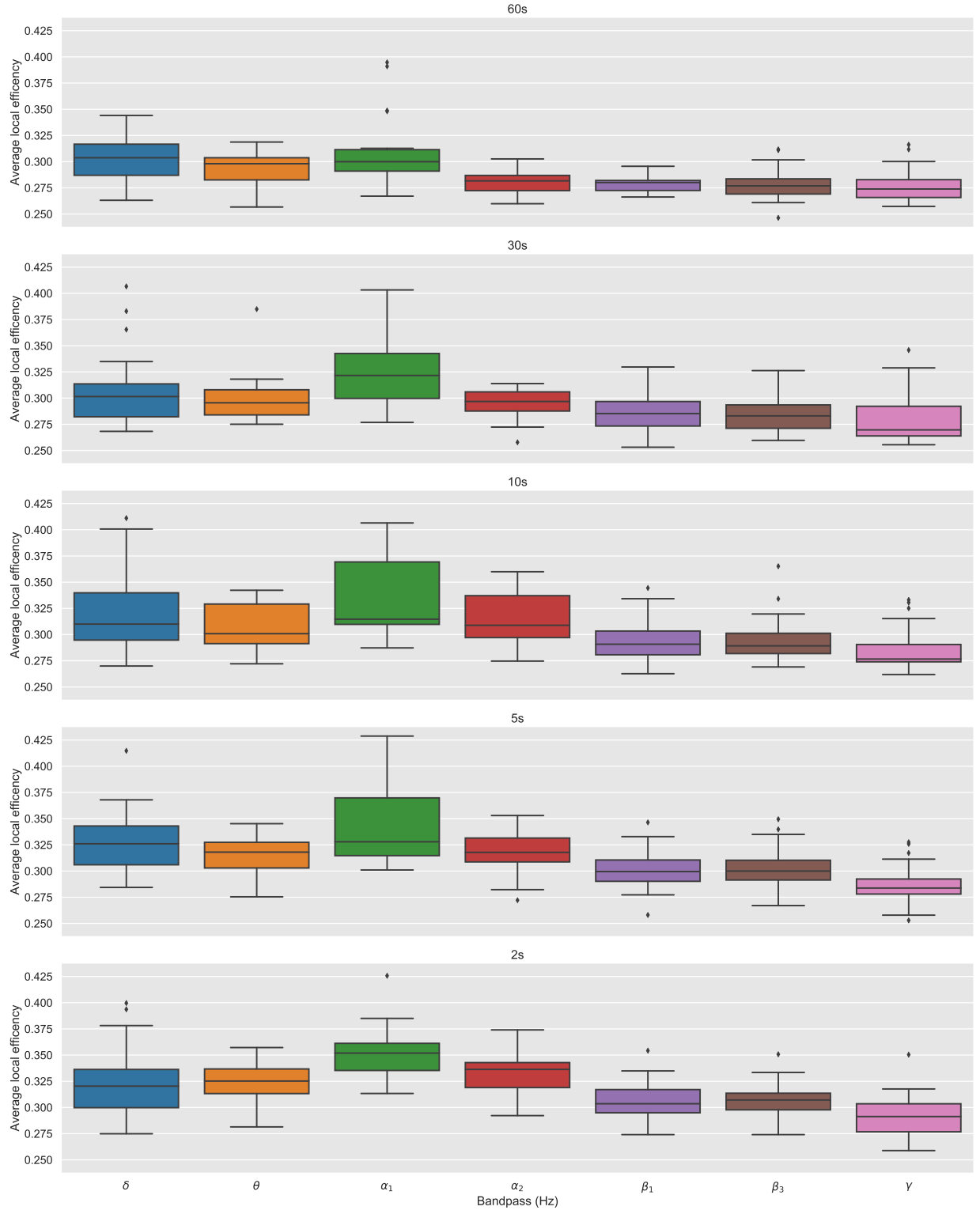


Figure 3.5: Average local efficiency per epoch length and bandpass.

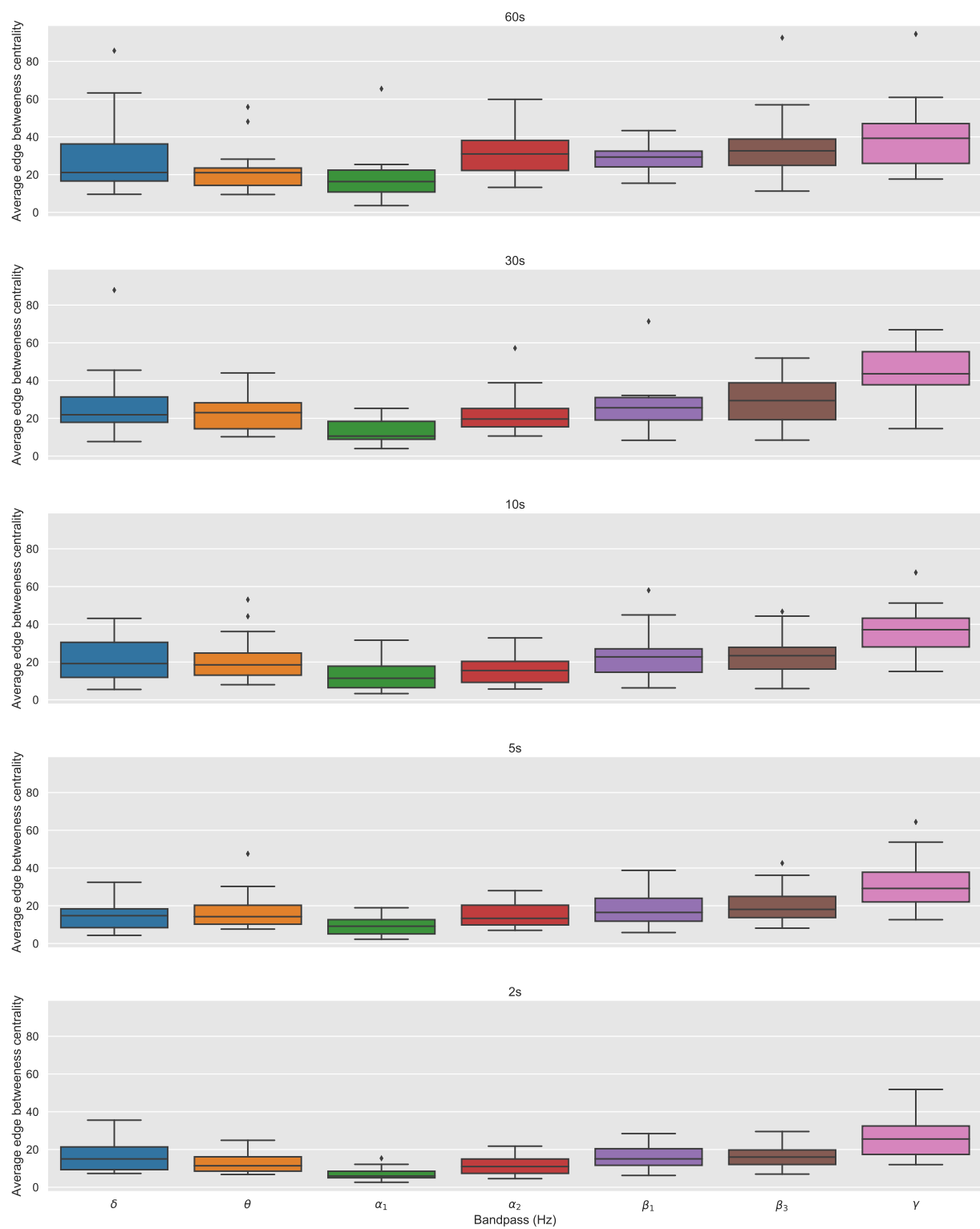


Figure 3.6: Average edge betweenness per epoch length and bandpass.

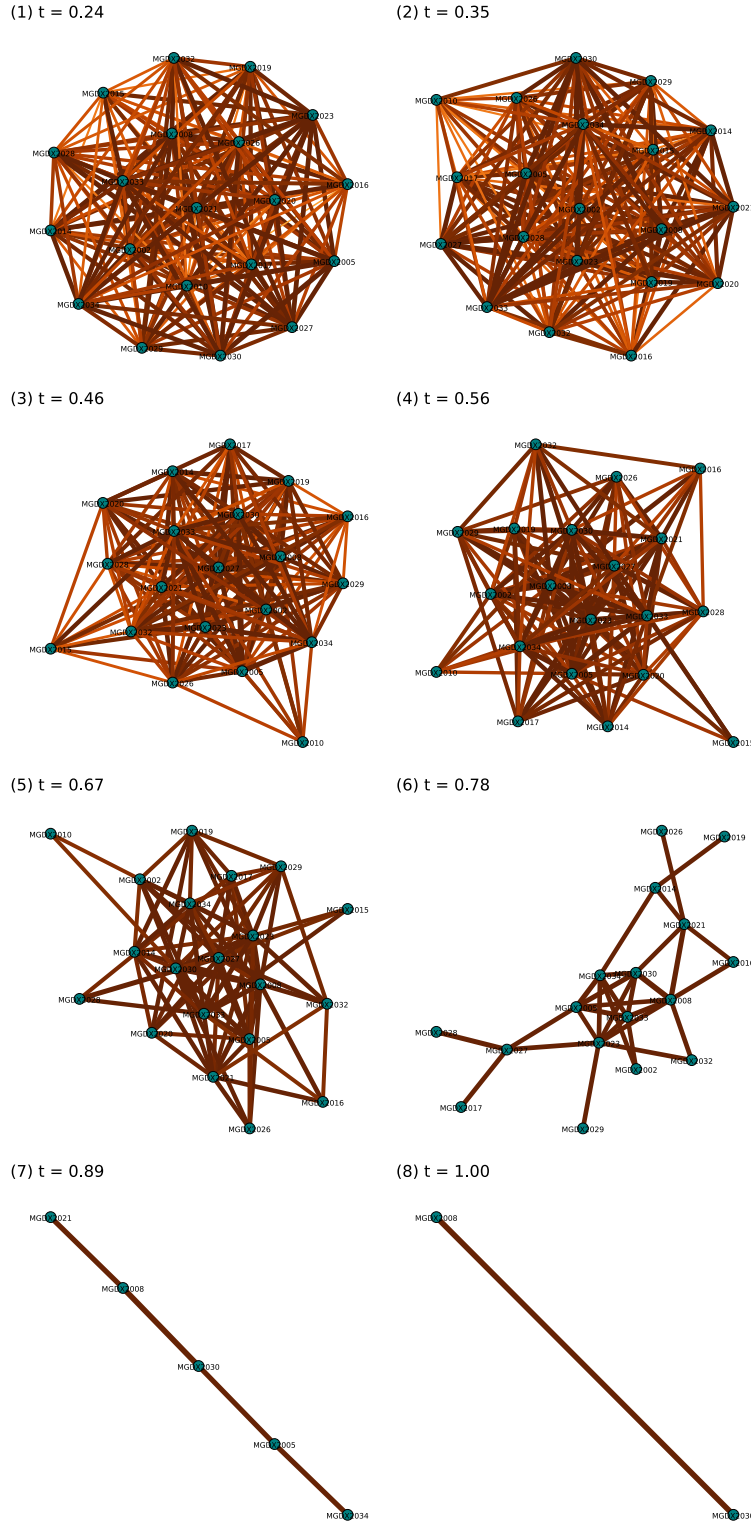


Figure 3.7: Euclidian based similarity between participants across graph metrics for a number of thresholds. Edge color and thickness represents the strength of association.

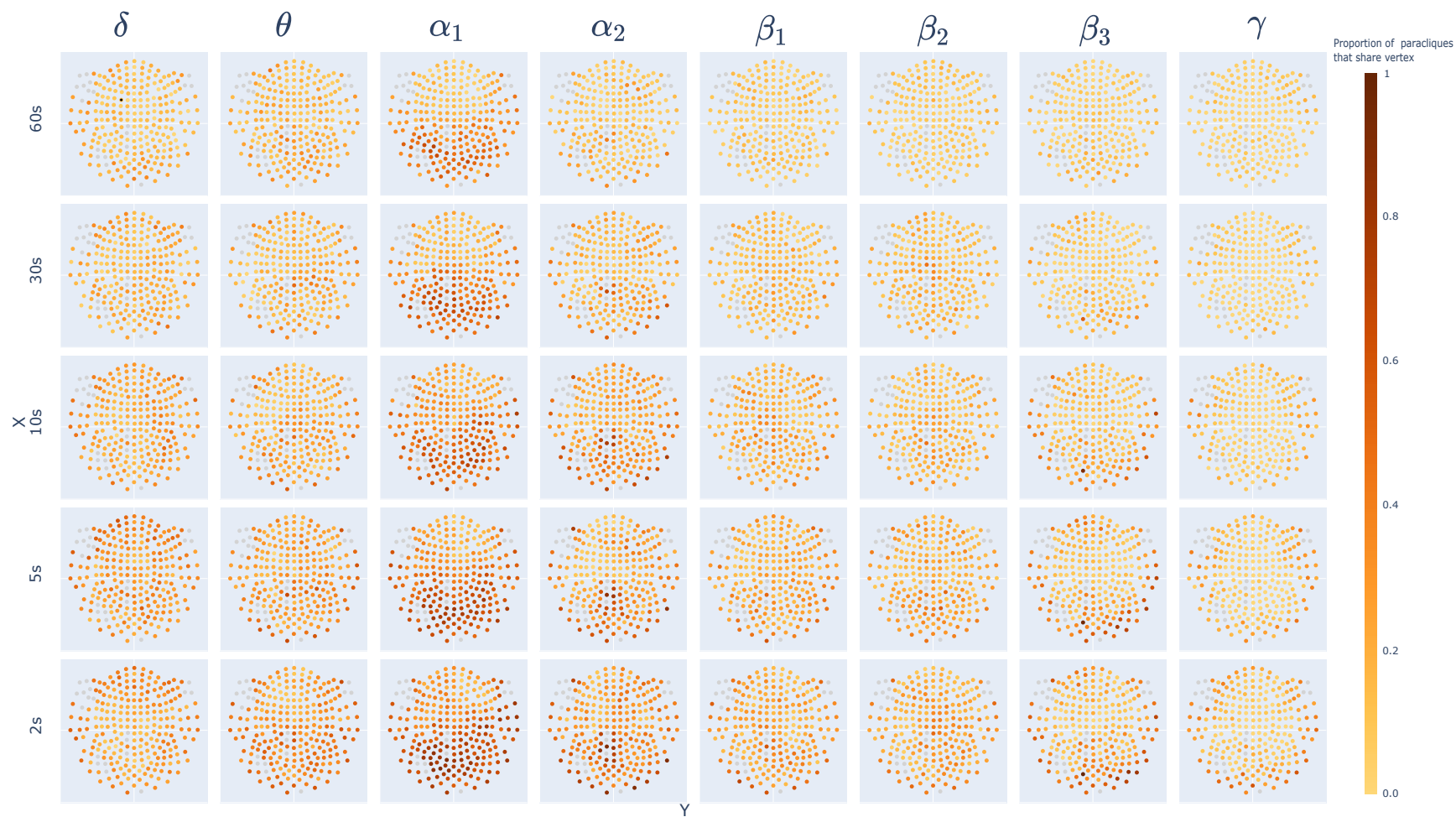


Figure 3.8: Top shared vertices across paracliques of individual MEG graphs for bandpasses and epoch lengths. X-axis is anterior (nose) to posterior, Y-axis is side to side.

however, in the α_1 bandpass it is evident that a large number of vertices are consistently present in clusters across individuals. Noteworthy too, is the consistency in bandpass β_2 that is focused in the sensors in the temporal and occipital regions.

The distribution of Jaccard similarity between graphs is shown in Figure 3.9.

The highest concordance between graphs based on shared edges across all epoch lengths occurs in the α_1 bandpass. This brain wave has been historically associated with an “idling” (or attention deficit) state [17], which given that these graphs are based on resting state is highly predictable. Figure 3.10 shows the top edges shared by the highest proportion of participants in each band wave across the different epoch lengths. In general, the shorter the epoch length, the more focused the edges are to the posterior of the head - the parietal and occipital regions. This is not consistent in the α_1 bandwidth, where occipital rhythms are considered abnormal [1]. In the α_1 bandpass, the majority of edges are instead focused in the left and right parietal regions, and in the shortest epoch of two seconds, only in the left parietal region. In all of the bandpasses, many cross-region edges occur.

3.4 Conclusion

By using graph metrics and whole graph comparisons we were able to identify outlier participants, as well as conserved graph characteristics and graph regions. We found graph metrics to be highly conserved for the majority of band-passes, with the highest dispersion of the graph metrics occurring in the α_1 bandpass. We found few associations across participants based on graph clustering, but did find highly conserved edges. Resting state data allowed for the exploration of behavior at various epoch lengths. These results indicate that in general, while graph structure and edge localization is conserved, differences at the vertex connectivity level do occur. We identified differences in the graphs at the bandpass and epoch length level - indicating differential brain activity levels and activity duration. The construction of multiple graphs and graph based comparisons can allow us to identify factors that differentiate experimental groups.

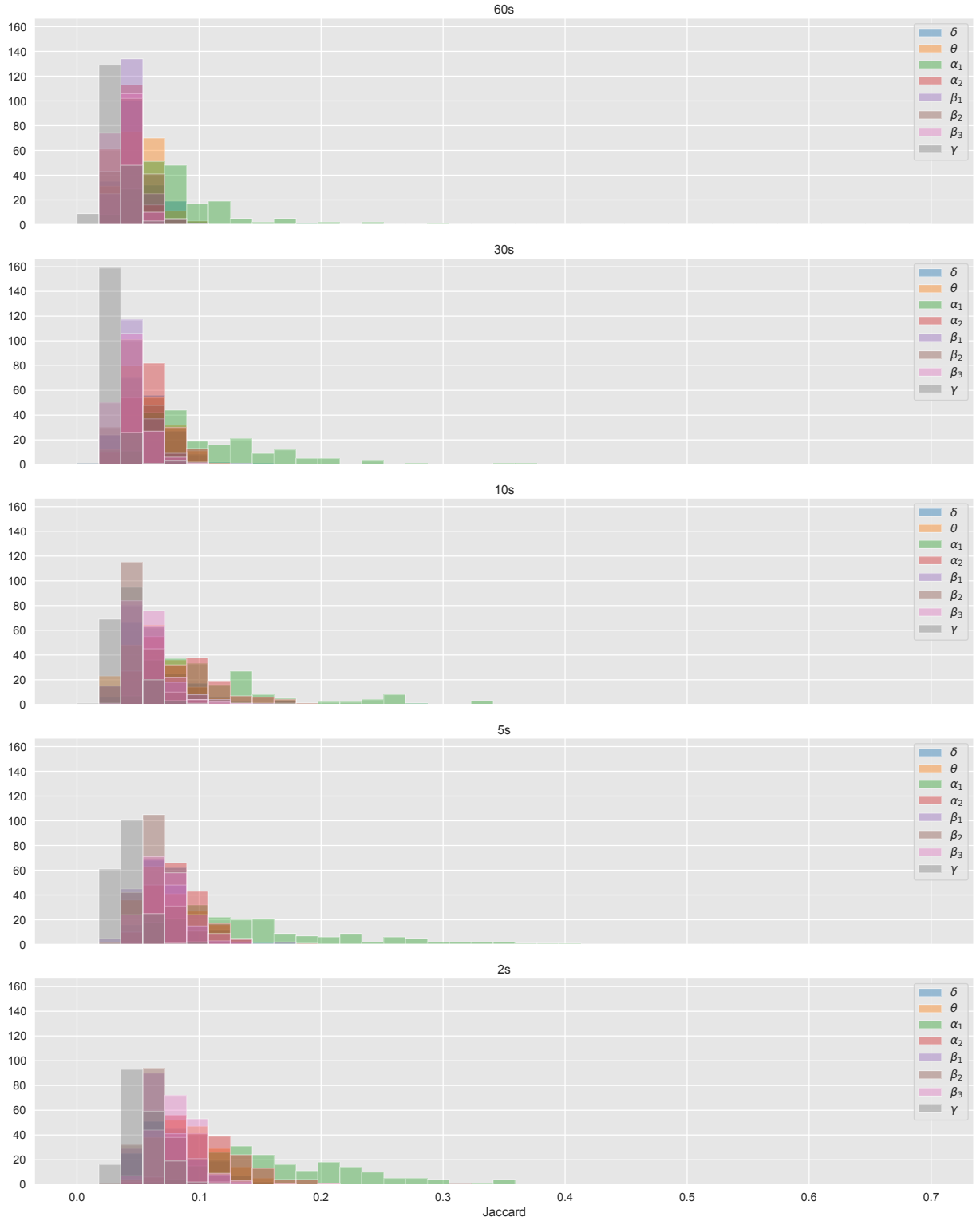


Figure 3.9: Jaccard distributions per epoch, separated by bandpass.

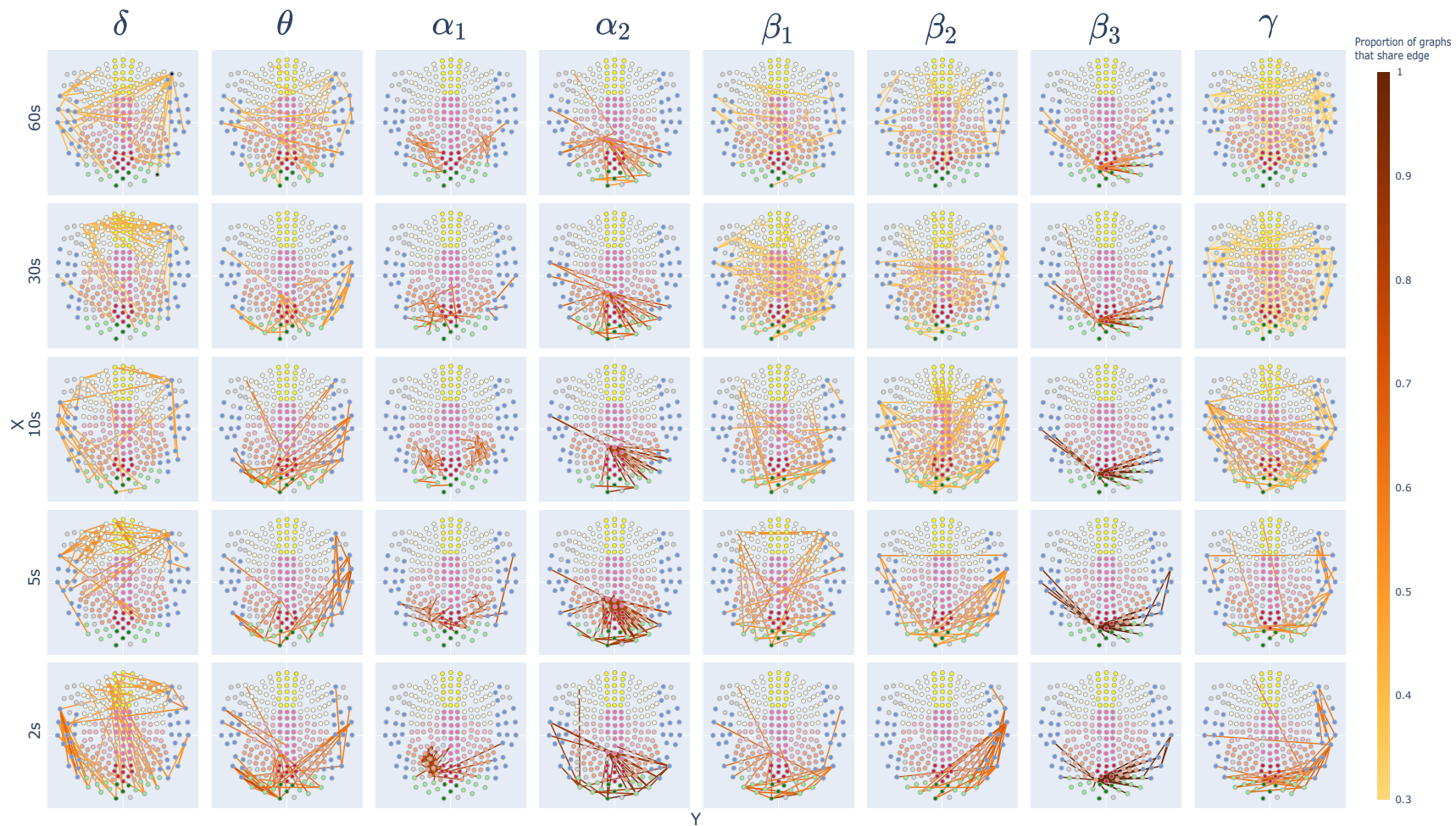


Figure 3.10: Top shared edge graphs for bandpasses and epoch lengths. X-axis is anterior (nose) to posterior, Y-axis is side to side.

3.5 Acknowledgements

We acknowledge Dr. Roozbeh who provided guidance, participant MEG profiles, performed ICA, and provided relevant background to resting state brain networks. We gratefully acknowledge the participant retention by CANDLE staff and particularly the parents and children who participated in this study. We also thank the Le Bonheur Children's Hospital Neuroscience Institute staff and the Radiological Technicians for their support.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

Bibliography

- [1] Babiloni, C., Del Percio, C., Boccardi, M., Lizio, R., Lopez, S., Carducci, F., Marzano, N., Soricelli, A., Ferri, R., Triggiani, A. I., Prestia, A., Salinari, S., Rasser, P. E., Basar, E., Famà, F., Nobili, F., Yener, G., Emek-Sava, D. D., Gesualdo, L., Mundi, C., Thompson, P. M., Rossini, P. M., and Frisoni, G. B. (2015). Occipital sources of resting-state alpha rhythms are related to local gray matter density in subjects with amnesic mild cognitive impairment and Alzheimer's disease. *Neurobiology of Aging*, 36(2):556–570. [74](#)
- [2] Boly, M., Phillips, C., Tshibanda, L., Vanhaudenhuyse, A., Schabus, M., Dang-Vu, T. T., Moonen, G., Hustinx, R., Maquet, P., and Laureys, S. (2008). Intrinsic brain activity in altered states of consciousness: how conscious is the default mode of brain function? *Annals of the New York Academy of Sciences*, 1129:119. [61](#)
- [3] Buchanan, A., Wang, X., and Gollan, J. K. (2014). Resting-state functional connectivity in women with major depressive disorder. *Journal of psychiatric research*, 59:38–44. [62](#)
- [4] Buckner, R. L., Andrews-Hanna, J. R., and Schacter, D. L. (2008). The brain's default network: anatomy, function, and relevance to disease. *Annals of the New York Academy of Sciences*, 1124(1):1–38. [61](#)
- [5] Bullmore, E. T. and Bassett, D. S. (2011). Brain graphs: Graphical models of the human brain connectome. *Annual Review of Clinical Psychology*, 7(1):113–140. PMID: 21128784. [66](#)
- [6] Chesler, E. J. and Langston, M. A. (2006). *Combinatorial Genetic Regulatory Network Analysis Tools for High Throughput Transcriptomic Data*, volume 4023, pages 150–165. Springer. [68](#)
- [7] Connectome Coordination Facility (2014). Components of the human connectome project - meg and eeg - connectome. [viii](#), [61](#)
- [8] Dimitriadis, S. I., Laskaris, N. A., Simos, P. G., Micheloyannis, S., Fletcher, J. M., Rezaie, R., and Papanicolaou, A. C. (2013a). Altered temporal correlations in resting-state connectivity fluctuations in children with reading difficulties detected via meg. *Neuroimage*, 83:307–317. [64](#)
- [9] Dimitriadis, S. I., Laskaris, N. A., Simos, P. G., Micheloyannis, S., Fletcher, J. M., Rezaie, R., and Papanicolaou, A. C. (2013b). Altered temporal correlations in resting-state connectivity

fluctuations in children with reading difficulties detected via MEG. *NeuroImage*, 83:307–317.

64

- [10] Escudero, J., Hornero, R., Abásolo, D., Fernández, A., and López-Coronado, M. (2007). Artifact removal in magnetoencephalogram background activity with independent component analysis. *IEEE Transactions on Biomedical Engineering*, 54(11):1965–1973. 63
- [11] Gusnard, D. A. and Raichle, M. E. (2001). Searching for a baseline: functional imaging and the resting human brain. *Nature reviews neuroscience*, 2(10):685. 61
- [12] Hagan, R. D., Langston, M. A., and Wang, K. (2016). Lower bounds on paraclique density. *Discrete Applied Mathematics*, 204:208–212. 68
- [13] Liu, Z., Fukunaga, M., de Zwart, J. A., and Duyn, J. H. (2010). Large-scale spontaneous fluctuations and correlations in brain electrical activity observed with magnetoencephalography. *Neuroimage*, 51(1):102–111. 66
- [14] Mantini, D., Penna, S. D., Marzetti, L., De Pasquale, F., Pizzella, V., Corbetta, M., and Romani, G. L. (2011). A signal-processing pipeline for magnetoencephalography resting-state networks. *Brain Connectivity*, 1(1):49–59. 63
- [15] Nomi, J. S. and Uddin, L. Q. (2015). Developmental changes in large-scale network connectivity in autism. *NeuroImage: Clinical*, 7:732–741. 62
- [16] Ortiz, E., Stingl, K., Münssinger, J., Braun, C., Preissl, H., and Belardinelli, P. (2012). Weighted phase lag index and graph analysis: preliminary investigation of functional connectivity during resting state in children. *Computational and mathematical methods in medicine*, 2012:186353. 64
- [17] Pfurtscheller, G. (1992). Event-related synchronization (ERS): an electrophysiological correlate of cortical areas at rest. *Electroencephalography and Clinical Neurophysiology*, 83(1):62–69. 74
- [18] Shu, I.-W., Onton, J. A., Prabhakar, N., O’Connell, R. M., Simmons, A. N., and Matthews, S. C. (2014). Combat veterans with ptsd after mild tbi exhibit greater erps from posterior–medial cortical areas while appraising facial features. *Journal of affective disorders*, 155:234–240. 62

- [19] Sporns, O. and Zwi, J. D. (2004). The small world of the cerebral cortex. *Neuroinformatics*, 2(2):145–162. [62](#)
- [20] Stam, C. J. (2005). Nonlinear dynamical analysis of eeg and meg: review of an emerging field. *Clinical neurophysiology*, 116(10):2266–2301. [62](#)
- [21] Thornburgh, C. L., Narayana, S., Rezaie, R., Bydlinski, B. N., Tylavsky, F. A., Papanicolaou, A. C., Choudhri, A. F., and Völgyi, E. (2017). Concordance of the resting state networks in typically developing, 6-to 7-year-old children and healthy adults. *Frontiers in human neuroscience*, 11:199. [62](#)
- [22] Vanhaudenhuyse, A., Demertzi, A., Schabus, M., Noirhomme, Q., Bredart, S., Boly, M., Phillips, C., Soddu, A., Luxen, A., Moonen, G., et al. (2011). Two distinct neuronal networks mediate the awareness of environment and of self. *Journal of cognitive neuroscience*, 23(3):570–578. [61](#)
- [23] Vinck, M., Oostenveld, R., Van Wingerden, M., Battaglia, F., and Pennartz, C. M. (2011). An improved index of phase-synchronization for electrophysiological data in the presence of volume-conduction, noise and sample-size bias. *NeuroImage*, 55(4):1548–1565. [64](#)
- [24] Welch, P. (1967). The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73. [63](#)
- [25] Wu, L., Calhoun, V. D., Jung, R. E., and Caprihan, A. (2015). Connectivity-based whole brain dual parcellation by group ica reveals tract structures and decreased connectivity in schizophrenia. *Human brain mapping*, 36(11):4681–4701. [62](#)

Chapter 4

Metabolic Comparison of *Escherichia coli* Phylotypes

Abstract

This work is an analysis of *Escherichia coli* (*E. coli*) phylotypes in metabolic space. *E. coli* is one of the most widely studied organisms, and many pathogenic strains are causes of disease. There are several known major subdivisions of *E. coli*, known as “phylotypes.” *E. coli* phylotypes often correspond to the both habitat (environment) and pathology of the strains, and are therefore of interest in the distribution of pathogenic strains. Using over 2,000 *E. coli* genomes, we developed a new, data-driven method for determining the core proteomes of 14 *E. coli* phylotypes. From the core proteomes we identified their core metabolism. We then compared the predicted metabolic networks of these phylotypes. These results have implications for distinguishing pathotypes, the metabolism of *E. coli* pathology, and human health and agriculture.

4.1 Introduction

Escherichia coli (*E. coli*) has important research and health implications. It is a Gram-negative bacterium, commonly found in the environment and in the microbiome of animals, including humans. Many strains are harmless, living commensally with their hosts. A number of pathogenic strains, however, are the cause of a number of human and animal host diseases. These include diarrhea, urinary tract infections, pneumonia, meningitis, and septicemia [2, 13].

A phylotype is a grouping of species or subspecies based on sequence similarity, and may represent an organism’s evolution and speciation events. Based on the methodology used, between four and seven *E. coli* phylotypes have previously been identified [8, 11]. In the past, phylotyping has been performed via PCR and looking for the presence or absence of specific gene presence or absence. With the advent of high throughput sequencing, much of phylotyping is done using genetic and sequence similarity. As more sequenced genomes and novel methods for classifying *E. coli* have become available, phylotype clustering is providing more refined groupings.

4.1.1 Historic *E. coli* population structure

Structure of the *E. coli* population has been studied using three different approaches to subtyping: (1) pathotype - based on resultant host disease, (2) serotype (or serovar) - based on cell surface

antigens, and (3) genotype (or biotype, or phylotype) - based on genetic composition. Serotype studies pre-date the era of molecular biology, and [22] was one of the first studies to associate specific serotypes with diarrheagenic diseases. It has become apparent however, that because of recombination and “hitch-hiker” variants, serotypes are not always reflective of the genetic diversity of phylotypes [6]

The standard reference strains of *E. coli* (ECOR) consists of 72 strains, purposely collected from a diverse range of hosts and locations was established in 1984 by Ochman and Selander [20]. The majority of subsequent phylogenetic studies have been based on this dataset [6]. For example, six groups (A, B1, B2, C, D, and E) were identified based on enzyme electrophoresis from 35 loci. A similar study, based on 38 loci, found evidence for 4 major subgroups (A, B1, B2, and D) as well as unclassified genomes placed in a group E [6]. Later studies were able to utilize nucleotide sequences of specific genes to build phylogenetic trees. Again, these are based on specific (and limited) loci within the *E. coli* chromosome.

The most commonly referred to phylotypes are: A, B1, B2, D, E, and F. Figure 4.1 shows previously defined relationships between these phylotypes based on iterative PCR of 13 gene sequences [7]. This resulted in the seven previously defined phylotypes, and a cryptic group Clade I. These and other studies have shown that the identified phylotypes have distinct properties, and play distinct roles. Indeed different phylogroups are more commonly detected in different species, as well as in the same species but in different geographic locations (reviewed in [28]). In a domestic animal *E. coli* host study, Morcatti Coura et al. [18] found phylotypes A and F associated with poultry, B1 and E associated with cattle and B2 and D associated with water buffalo.

Since it has been found that *E. coli* phylotypes often correspond to the environment and pathology of the strains, phylotypes are of interest in the distribution of pathogenic strains [4].

4.1.2 Recent developments in *E. coli* phylogeny

In the largest such study to date [1], MASH genomic sequence distances [21], pairwise Average Nucleotide Identity (ANI) scores, and hierarchical clustering were used to identify 14 distinct phylotypes from over 100,000 *E. coli* genomes. These genomes were found to be distributed in the following phylotypes: A, B1, B2-1, B2-2, C, D1, D2, D3, E, F1, F2, O157, Shig1, and Shig2.

Figure 4.2 shows the population structure that they found. Instead of being based on specific, limited loci sequences or genes, this study is based on whole genome sequence similarity, and therefore considers the entire genome diversity, so-called phylogeomics, instead of phylogenetics.

4.1.3 Similar work in *E. coli* phylogenetic metabolism

To compare subgroups, we often utilize the *core* genes, *pan* genomes, and *unique* genes. The core genes of a subset of organisms (potentially a species or a set of similar strains) are the genes that are present in all (or nearly all) members of the subset. Equivalently, the pan genome consists of all genes present across the subset of organisms. The unique genes are those only found in a single organism in the set. For completeness, the *accessory genome* contains the genes that are neither in the core or the unique genes (see Figure 4.3).

A 2011 study computed the metabolic networks of 29 strains of *E. coli*, which included strains from each of the historic A, B1, B2, D, E, and F groups [30]. A number of the genomes of these strains had previously been annotated, and were re-annotated using MICheck. The annotated proteins were then imported and homogenized using BioCyc Pathway Tools ¹, and EcoCyc as the reference annotation set. OrthoMCL [16] was then used to calculate the core and pan genome of the 29 strains. Independently, the core metabolism was defined as reaction annotations present in all strains.

In contrast to [30], this study considers *E. coli* phylogeny based on genome-wide differences and a far larger sample size (100,00 instead of 29). Similarly, the phylotype specific metabolisms are also based on a larger dataset with up to 200 genomes per phylotype instead of 29 in total. When determining phylotype core and pan genomes, each phylotype is treated independently, preventing outlier phlotypes from affecting individual results. During the course of the study, significant issues with BioCyc and MetaCyc protein annotations were found. For consistency in the final comparisons, the starting material was the DNA sequence for each genome, which was then run through Prokka [27] for prediction of protein encoding sequences, and annotated using EggNOG

¹BioCyc is a collection of over 100,000 databases, termed *Pathway/Genome Databases* (PGDB) [14]. Each PGDB corresponds to an organism model, based on a genome. A PGDB contains genes, proteins, enzymes, reactions and metabolic pathways for the organism, amongst other annotations. EcoCyc is the PGDB for *E. coli* K-12 MG1566, and MetaCyc is a PGDB database for all the reactions, compounds and pathways that occur across BioCyc [5].

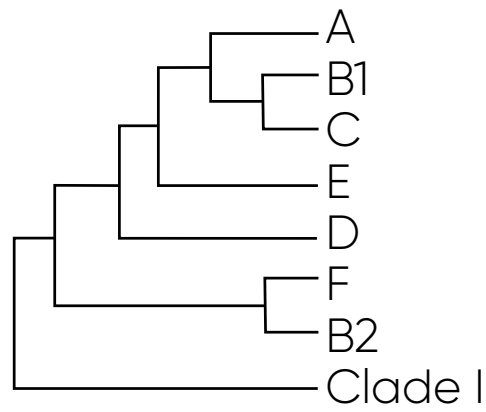


Figure 4.1: A past example of phylogenetic relationships between *E. coli* phylotypes. Based on Figure 1 of [7].

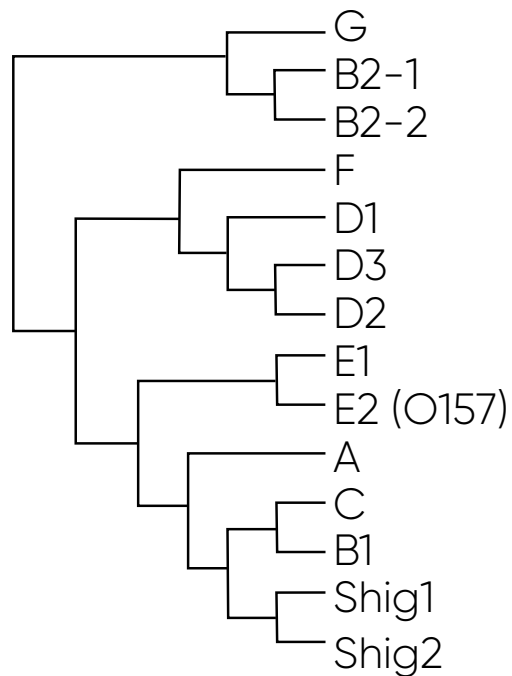


Figure 4.2: *E. coli* phylotype population structure according to [1].

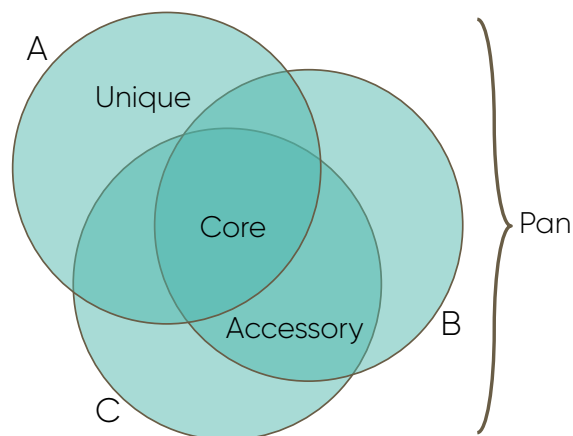


Figure 4.3: Genome definitions for hypothetical strains A, B, and C.

[10]. Finally, the core metabolism for the phylotypes are based on the core set of proteins, not from independently collected annotations.

4.1.4 Aims and approach

When referring to phylotype groupings in the rest of this text, unless otherwise stated, we will be referring to the phylotypes corresponding to those in [1], as visualized in Figure 4.2. Currently our understanding of how *E. coli* phylotypes differ at the functional level remains incomplete, in part because previous work has not provided an in-depth and complete comparison of metabolic networks across a large dataset. This study aims to use these 14 phylogenetic groupings to identify functional differences at the protein and metabolic levels. With this aim, we develop a systematic method for finding cut-off thresholds for which genes to include in core genomes, and compare core genomes at the protein and metabolic levels.

Our interest lies with the unique genes in the phylotype-specific cores. Characterization of phylotype specific factors may help in understanding the transition between pathogenic bacteria and commensal bacteria. We also seek to understand the degree to which pathogenicity is associated with phylotypes, and therefore how it may have evolved. Factors that are associated with host and environmental specificity may also play a role.

4.2 Methods

In short, we use two datasets (BioCyc and GenBank) to find the core genomes of each phylotype. Using the core genomes, we derive the core metabolic networks of the phylotypes. We then compare these networks to identify unique features of the phylotypes. Figure 4.4 is a succinct overview of the approach per phylotype.

4.2.1 Data acquisition

Assembled genomes were downloaded from BioCyc and Genbank. For each of the assemblies, genome quality was calculated as in [31]. A genome quality score is a value between 0 and 1, and is based on assembly quality and completeness. We considered only the genomes with a score at or above the author’s “safe-to-use” value of 0.80. This was done to ensure only genomes with high quality sequencing and assembly contributed toward our results.

The phylotype label for each of the genomes was defined as in [1].

BioCyc

All *E. coli* PGDBs available from BioCyc on 14/11/2018 were downloaded (using <http://brg-files.ai.sri.com/subscription/dist/flatfiles-52983746/index.html>). This produced a total of 551 PGDBs. After filtering and labeling by phylotype, this resulted in a total of 476 genomes. Of these, Table 4.1 shows the number of genomes per phylotype. These PGDBs contain both the assembled genome and annotated proteins. We only considered the phylotypes with at least 20 members.

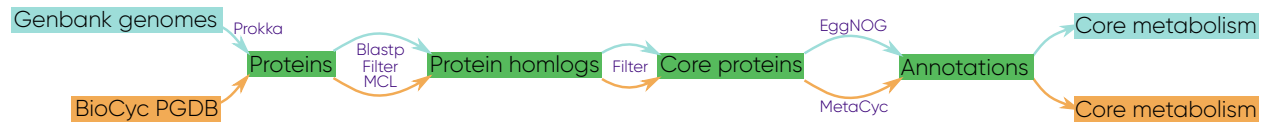


Figure 4.4: Outline of method for each phylotype.

Genbank

Previously phylotyped *E. coli* genomes were downloaded from NCBI's Genbank API. Up to 200 genomes per phylotype were acquired. Proteins were called using Prokka [27].

4.2.2 Core genome pipeline

Protein clusters

Following similar methodology to OrthoMCL [16] and Roary [23], we based the core genome calculations on protein sequence similarity and clustering. Pairwise sequence similarity for each pair of protein sequences across and within genomes was found using BLASTp [3] in a parallel fashion. We used a cutoff of 10^{-6} for the e-value and 95% sequence identity. For each phylotype we combined the BLASTp results into an edge weighted graph, and ran MCL [29] with the bitscore as the weight between pairs of sequences. Each MCL cluster was treated as a homologous protein cluster.

Core genome calculation

The core proteins are based on selecting the protein clusters that contain at least some percent of the genomes under consideration. A “hard core” is based on setting this percentage at 100%, and a “soft core” at some arbitrary cutoff of less than 100%. Gordienko et al. [9] defines the soft core cutoff as 85%, while Kaas et al. [12] uses 95%. In noisy, real world data a cutoff of 100% may result in false negatives and even in an empty core, while choosing a cutoff for a soft core is arbitrary with little biological basis. The hard core cutoff also results in a direct dependency between the number of genomes considered, and the resultant core genome size, as seen in Fig. 4.5(A). For this figure, at each point we randomly selected the genomes, and calculated the core sizes only considering these genomes.

To study further the relationship between the cutoff and the core size, we calculated the data points (x, y) for $x = 1, 2, \dots, N$, where N is the total number of genomes and y = the number of protein clusters that contain proteins from at least x genomes. We found the shape of this curve to be consistent across the phylotypes, as well as when ignoring phylotypes and considering all genomes in the BioCyc dataset. In general, we can explain the shape of the curve as follows: the

Table 4.1: Number of member genomes selected from NCBI per phylotype.

Phylotype	Number of NCBI genomes	Number of BioCyc genomes
A	200	88
B1	200	150
B2-1	200	101 (B2-1 and B2-2 combined as B2)
B2-2	200	-
C	200	-
D1	200	43
D2	177	-
D3	177	-
E	200	94
F1	96	-
F2	199	-
O157	200	-
Shig1	177	-
Shig2	200	-
Total	2626	476

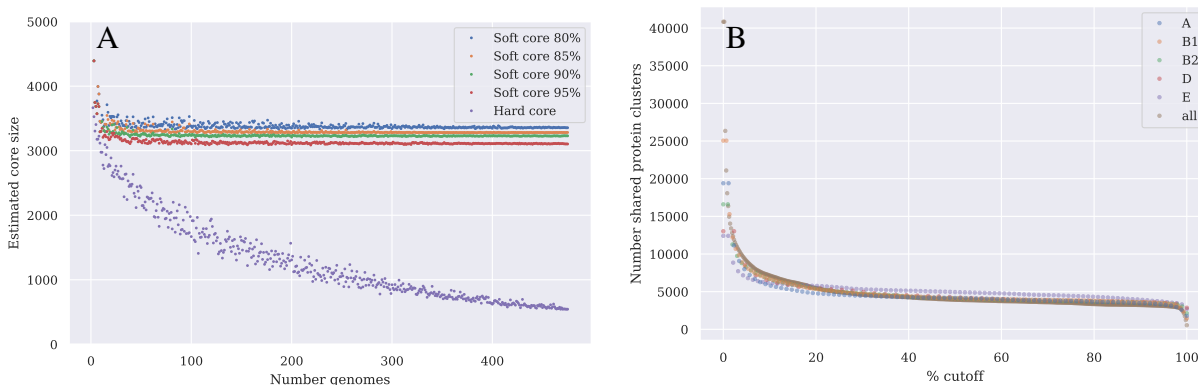


Figure 4.5: Analysis of the core genome and development of core genome cut-off estimate: (A) Core genome sizes across a random selection of genomes, for the hard core and for several soft core estimates. (B) Core genome sizes per phylotype, for range of cutoff percentages.

first data point contains all the given protein clusters (i.e. the pan-genome); there is then a sharp drop-off, until a plateau is reached; this plateau continues, until the core size drops sharply again. This pattern for the genomes downloaded from BioCyc is shown in Fig. 4.5(B). The shape of the curve is also very consistent, even though the number of genomes in each group is different.

We hypothesize that the initial sharp drop-off is the removal of non-core genes (the pan and accessory genes), and the second sharp drop-off occurs due to vanishing data - i.e. missing genes due to noisy sequencing and assemblies. We therefore propose that a core genome cutoff value should be at the “knee point” –the point just before the second drop-off. This could also be considered the point of diminishing-returns for the cutoff value.

To identify the knee value in this curve, we follow a similar methodology to “kneedle” [26]. Kneedle was initially developed to identify trade-offs in computer system behavior. We modify it slightly for this application. The procedure is given in Algorithm 4.1. Firstly, we ignore repeated values at the beginning and end of the data curve, and find the straight line that ends at the last value, and starts at the lowest point such that the line does not intersect the curve. We then calculate the distance from this line to the curve for each value of x , and select the point that corresponds to the largest distance. Fig. 4.6 shows the result of this method for phylotype B1 from the NCBI genomes.

To compare our results to a fixed threshold, (as those in Fig. 4.5(A)), we also tested this method by randomly selecting a number of genomes and estimating the core size. Fig. 4.7(A) shows our method compared to a soft core of 95%. Our method is more sensitive to data, but is still more consistent than the hard core. Improvements to the algorithm may provide more robust results. Fig. 4.7(B) compares the relationship between the core size and the number of genomes considered for the hard core and the new estimated core. The estimated core sizes have a much tighter distribution in size (between 3,000 and 4,000) than does the hard (between 500 and 3,000).

4.2.3 Annotation of core genomes

Annotations for the core proteins derived from the BioCyc genomes are based on MetaCyc reactions, and we labeled all core proteins as such (see Appendix 4.A for method). For comparison,

Algorithm 4.1 Core Genome Estimator

```
1: function STRAIGHTLINE( $(x_1, y_1), (x_2, y_2)$ )
2:   return straight line that passes through  $(x_1, y_1)$  and  $(x_2, y_2)$ 
3: end function
4: procedure COREGENOMEESTIMATOR(ProteinClusters,  $N :=$  number genomes)
5:   for  $x_i = 1, \dots, N$  do
6:      $y_i =$  Number clusters in ProteinClusters with at least  $x_i$  genome members
7:   end for
8:    $s = 0, n = N$ 
9:   while  $y_s == y_{s+1}$  do
10:     $s = s + 1$ 
11:  end while
12:  while  $y_n == y_{n-1}$  do
13:     $n = n - 1$ 
14:  end while
15:   $L =$  STRAIGHTLINE( $(x_s, y_s), (x_n, y_n)$ )
16:  while  $L$  intersects  $y$  do
17:     $s = s + 1$ 
18:     $L =$  STRAIGHTLINE( $(x_s, y_s), (x_n, y_n)$ )
19:  end while
20:  for  $i = s, \dots, n$  do
21:     $d_i = y_i - L(x_i)$ 
22:  end for
23:   $k = \operatorname{argmax}_i d_i$ 
24:  return  $x_k, y_k$ 
25: end procedure
```

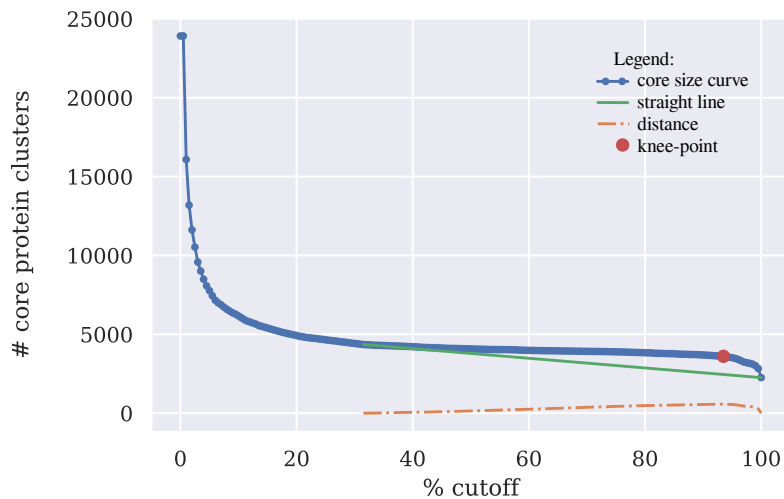


Figure 4.6: Core genome calculation for phylotype B1.

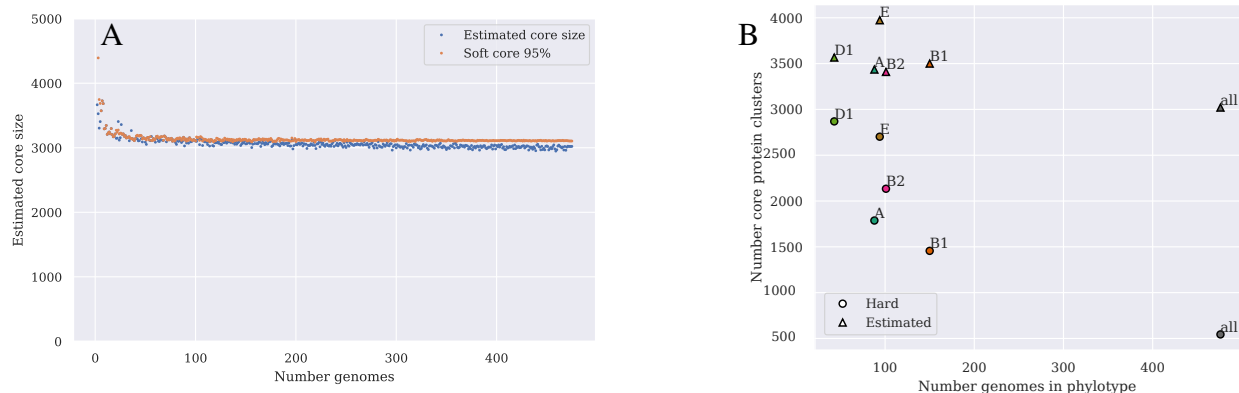


Figure 4.7: Analysis of the core genome and development of core genome cut-off estimate: (A) Estimated core genome sizes across random selections of BioCyc genomes using the new method, compared to using a soft core of 95%. (B) Comparison of core genome calculation and hard core size in BioCyc genomes.

we also annotated all proteins (from BioCyc and NCBI) with KEGG Orthologous (KO) terms using EggNOG [10].

Metabolic networks associated with core genomes

KO terms map to EC numbers and reactions in the KEGG database. Using these maps we were able to link protein clusters to reactions and, therefore, core protein clusters for each phylotype to core reactions for each phylotype. We then considered the metabolism of each phylotype associated with these core genomes.

For convenience, Neo4J [19] databases of the complete KEGG and BioCyc reaction and metabolic networks were created. The KEGG database is publicly available at <https://github.com/carissableker/neo4j-KEGG>. Using these databases we constructed core metabolic graphs for each phylotype. These are bipartite graphs consisting of the reactions associated with the core proteins, and their substrate and product metabolites (or compounds). The two sets in the graph are the set of reactions and the set of compounds (or metabolites). For comparison, we then combined the graphs by taking the union across phylotypes.

4.3 Results

Core genomes

The core genome sizes for the Biocyc dataset are shown in Table 4.2, and those for the NCBI genomes in Table 4.3. Previous work has shown the core *E. coli* genome to be between 2,200 [25] and 3,051 [12] genes. Our estimate of 3,020 lies at the high end of these estimates. Unsurprisingly the phylotype specific cores are larger than the *E. coli* core genome. All cut-off percentages were found to be between 88% and 98%, within the range of previously used soft core thresholds.

With the BioCyc genomes we found phylotype E to have the largest core genome and phylotype B2 to have the smallest, while the NCBI data gave C the largest core, and phylotype E the smallest. This result shows how sensitive genome sizes are to the initial data. Abram et al. [1] found phylotype E to have a significantly smaller core genome size compared to the other genomes, lending more credibility to the NCBI data pipeline.

Core genome annotations

After comparing the MetaCyc to the EggNOG annotations we noticed the MetaCyc annotations had significantly less agreement per protein cluster (see Fig 4.8). Annotations consistency within each cluster was starkly contrasted between the two approaches. In general, BioCyc annotated each protein within a cluster with different reactions, while the majority of EggNOG annotations within

Table 4.2: Core genome sizes based on BioCyc genomes

Phylotype	Number genomes	Hard core	Cutoff (%)	Core
A	88	1,787	92.05	3,435
B1	150	1,456	94.67	3,500
B2	101	2,134	96.04	3,409
D1	43	2,869	93.02	3,566
E	94	2,702	89.36	3,973
-	476	544	96.85	3,020

Table 4.3: Core genome sizes based on NCBI genomes.

Phylotype	Pan genome	Hard core	Cutoff (%)	Core
A	19,296	2,561	92.50	3,626
B1	23,912	2,255	93.50	3,606
B2 -1	13,520	3,189	96.50	3,747
B2-2	17,177	2,853	97.50	3,575
C	13, 820	2,882	96.50	3,914
D1	18,837	2,762	97.50	3,689
D2	24,398	2,694	94.92	3,598
D3	16,089	2,548	96.61	3,776
E	25,145	2,361	97.50	3,079
F1	14,093	3,155	94.79	3,890
F2	18,705	2,712	96.48	3,499
O157	9,749	3,537	95.00	4,278
Shig1	8,698	2,204	90.40	3,695
Shig2	7,932	2,911	94.50	3,981

a cluster was in agreement. Since EggNOG resulted in more consistent annotations, we decided to discard the BioCyc genome models and concentrated the rest of the analysis on the genomes based on assemblies downloaded from NCBI Genbank.

Mapping the core protein clusters to KEGG KO terms, and thereby KEGG reactions, resulted in a list of core-associated reactions for each phylotype. Table 4.4 contains the number of KO and reaction terms for each phylotype. 1,817 reactions were found to be common to all phylotypes.

Phylotypes B1, E, and F2 had no unique KO terms (and therefore no unique reactions). Phylotype O157 was found to have the highest number of unique KO terms and reactions, with phylotypes Shig1 and Shig2 following. The diversity in reactions presence and absence across the phylotypes can be seen in the clustered heatmap in Figure 4.9. To determine similarity between phylotypes, we used Ward clustering, after removing reactions that occur in all phylotypes. We can compare the dendrogram at the top of this figure to that in Figure 4.2. While a number of sister groups are consistent, (B2-1 and B2-2, B1 and C, Shig1 and Shig2) the overall structure has changed. Now three high-level branches appear, that of E, B2-1 and B2-2, a second branch containing Shig1 and Shig2, and a third branch of the remaining phylotypes.

Host source

Using the NCBI eutils API, we attempted to determine the host and disease of the source sample for each genome. Unfortunately, the majority had not been reported (see Appendix 4.C). Those that do are visualized in a heatmap in Figure 4.10.

Each of the phylotypes apart from D1 have appeared associated with diarrhea symptoms. B2-1 and B2-2 are both mostly associated with bacteremia (23/44, 17/29) and urinary infections (13/144, 8/29). C is associated with hemolytic uremic syndrome (hus, 9/13). D1 and D3 associates with bacteremia (31/42, 28/35). F1 is uniquely associated with cellulitis, a skin infection. Shig1 is associated with shigellosis (43/102) and diarrhea (36/102). Due to the limited amount of data available, these results cannot be determined to be significant, however, a Fisher Exact test of the contingency table gives a p -value of < 0.001 , suggesting disease type is significantly associated with phylotype.

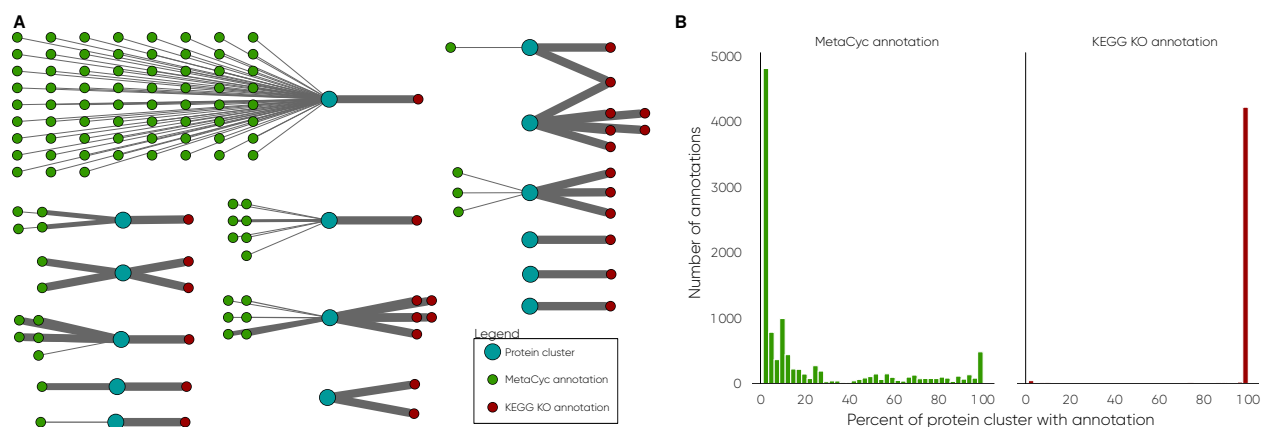


Figure 4.8: Comparison of MetaCyc and EggNog protein cluster annotations. (A) Visualization of annotations per cluster for a random selection of core clusters of phylotype A (BioCyc genomes). Blue vertices represent clusters, green vertices represent MetaCyc annotations, and red vertices represent KEGG KO terms. Link thickness represents the number of proteins in the cluster that are associated with the annotation. (B) Histogram of annotation consistency for phylotype B2 (BioCyc genomes). The left histogram shows the majority of MetaCyc annotations are not consistent within annotations (a low percentage indicates protein annotations within the cluster are not in agreement), and the right histogram shows the majority of EggNog KO annotations are highly consistent within clusters. Since disagreement is unexpected and undesirable in our analysis, we proceeded with only using the NCBI genomes and EggNOG annotations.

4.4 Discussion

Fig. 4.11 shows the complete graph of reactions and compounds associated with each of the phylotype core genomes (for visualization we do not show currency compounds in this figure). This graph allows us to identify reactions and compounds that are associated with a single phylotype. Such reactions linked to a single phylotype in this manner do not mean a strain in another phylotype does not have the capability for this reaction, because the corresponding proteins may appear in their accessory genomes. Rather, we can hypothesize that all strains within this phylotype will most likely have the ability of this reaction.

Though phylotype A contains no unique core reactions, it does have two KO terms: K13655 (antitoxin *mqsA*) and K13651 (motility quorum-sensing regulator *mqsR*). *mqsR* and *mqsA* proteins are a toxin/antitoxin (TA) pair of *E. coli* [15]. A toxin/antitoxin pair consists of a toxin (that may cause cell death) and a corresponding antitoxin that is able to bind to and neutralize the toxin. The *mqsR/mqsA* protein complex has been found to be a regulatory protein, regulating

Table 4.4: Core genome annotations per phylotype.

Phylotype	No. of core KO terms	No. of core reactions	No. of unique KO terms	No. of unique reactions
A	3176	2163	2	0
B1	3156	2147	0	0
B2-1	3202	2100	18	11
B2-2	3099	2070	1	0
C	3260	2182	21	9
D1	3202	2142	2	3
D2	3148	2124	1	0
D3	3218	2157	3	3
E	2829	1945	0	0
F1	3257	2185	6	2
F2	3079	2115	0	0
O157	3331	2187	41	24
Shig1	3124	2108	22	8
Shig2	3239	2167	30	20

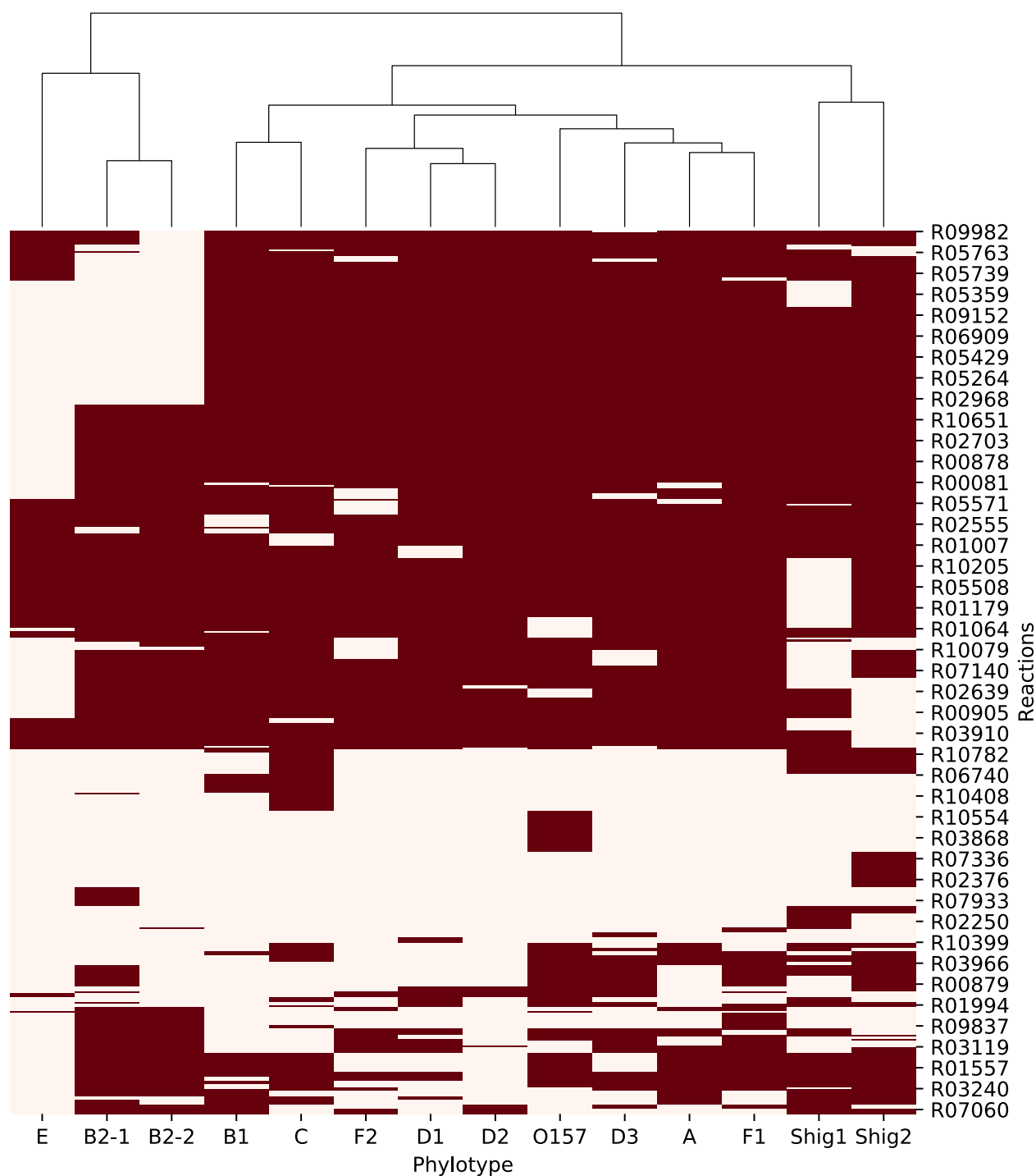


Figure 4.9: Phylotype clustering based on reaction presence in the core genome. Red indicates presence, and white indicates absence. Reactions present in all phylotype cores are not shown.

both itself and another toxin (cspD), among others [15]. The same authors propose this particular TA system is related to biofilm formation, quorum and cell density sensing and programmed cell death.

The majority (10 of the 18) of B2-1 unique KO terms occur in the biosynthesis of siderophore group nonribosomal peptides pathway. Additionally (9 out of 11) of B2-1 unique reactions occur in the biosynthesis of unsaturated fatty acids pathway. Siderophore presence has shown to be a factor in *E. coli* associated bacteremia [17], tying into the finding that the majority of host diseases on phylotype B2-1 samples are bacteremia.

The single unique KO term of B2-2 corresponds to the tlyA protein, a protein with haemolytic and RNA methylation activity originally identified in *Mycobacterium tuberculosis* where it is thought to contribute to its pathogenesis.

Among the unique reactions of phylotype C is a core steroid degradation reaction (oxidoreductase R09860). Phylotype C was found to be the phylotype most associated with HUS, and the inclusion of this reaction may tie-in to the finding that oral steroid treatment had no effect on HUS duration [24]. Other elements unique to phylotype C include four reactions (eight KO terms) in the phenylalanine metabolism pathway, and two reactions (six KO terms) involved in the pentose phosphate pathway.

Two KO terms are unique to D1. The first (corresponding to EC 1.1.1.353), maps to three reversible reactions, all of which are in the aflatoxin biosynthesis pathway. The second is an RNA-directed DNA polymerase. Aflatoxins are toxins produced by fungi, that can lead to hepatocellular carcinoma. Wang et al. [32] isolated an *E. coli* strain from chicken cecum that was able to eliminate aflatoxins.

D2 was found to only possess one unique KO term: K02247, related to the competence protein ComGE which is required for DNA transformation.

The three KO terms unique to D3 are K06044 (treY), K13285 (sipB) and K09144 (an uncharacterized protein). treY is an enzyme in starch and sucrose metabolism. The protein sipB is a cell invasion protein, related to salmonella infection, shigellosis, and bacterial invasion of epithelial cells.

Two of the KO terms unique to F1 are linked to deacylases not associated with any pathways. Unique to F1 is also the TA system of doc/phd which is part of the prokaryotic defense system (K07341, K19165). Thirdly is a rhomboid-like inter-membrane protease (K09650).

As previously stated, phylotype 0157 has the largest number of unique KO terms and reactions. These KO terms are included in the pathways of pathogenic *E. coli* infection, iron (III) ABC transporters, salmonella infection, bacterial secretion system, shigellosis, biofilm formation, and bacterial invasion of epithelial cells among others. The unique reactions occur in butanoate metabolism (five reactions), naphthalene degradation (four reactions), aromatic compound degradation, and antibiotic biosynthesis, among others.

Shig1 unique KO terms include those involved in quorum sensing, biofilm formation, streptomycin biosynthesis, methane metabolism and antibiotic biosynthesis. These reactions occur in glycerolipid metabolism, streptomycin biosynthesis, and biosynthesis of antibiotics.

Six Shig2 unique KO terms occur in propanoate metabolism (two reactions), five (five reactions) occur in porphyrin and chlorophyll metabolism. Shig2 unique reactions also include penicillin amidohydrolase.

The Shiga-like toxin 1 subunits stxB and stxA are not present in any of the core genomes, suggesting these are accessory genes. Of the 152 KO terms in the Pathogenic *E. coli* infection pathway (ko05130) eight are present in at least one of the core genomes. All eight are present in phylotype 0157.

4.5 Conclusion

We have developed a new, data-driven method for determining a core genome threshold. Using core genomes per phylotype we were able to determine phylotype specific molecular functions and reactions in *E. coli*. The pipeline described here can easily be extended as new data becomes available and can be applied to other organisms. The results may have implications for distinguishing pathotypes, understanding the metabolism of *E. coli* pathology, and improving human health and agriculture.

4.6 Acknowledgements

This work was done in collaboration with Professor David Ussery, Dr Zulema Dominguez, Visanu Wanchai and Kaleb Abram of the Arkansas Center for Genomic and Epidemiology Medicine at University of Arkansas for Medical Sciences.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

Funding also came from The Helen Adams & Arkansas Research Alliance Professor & Chair and from NIH/NIGMS (grant number 1P20GM121293).

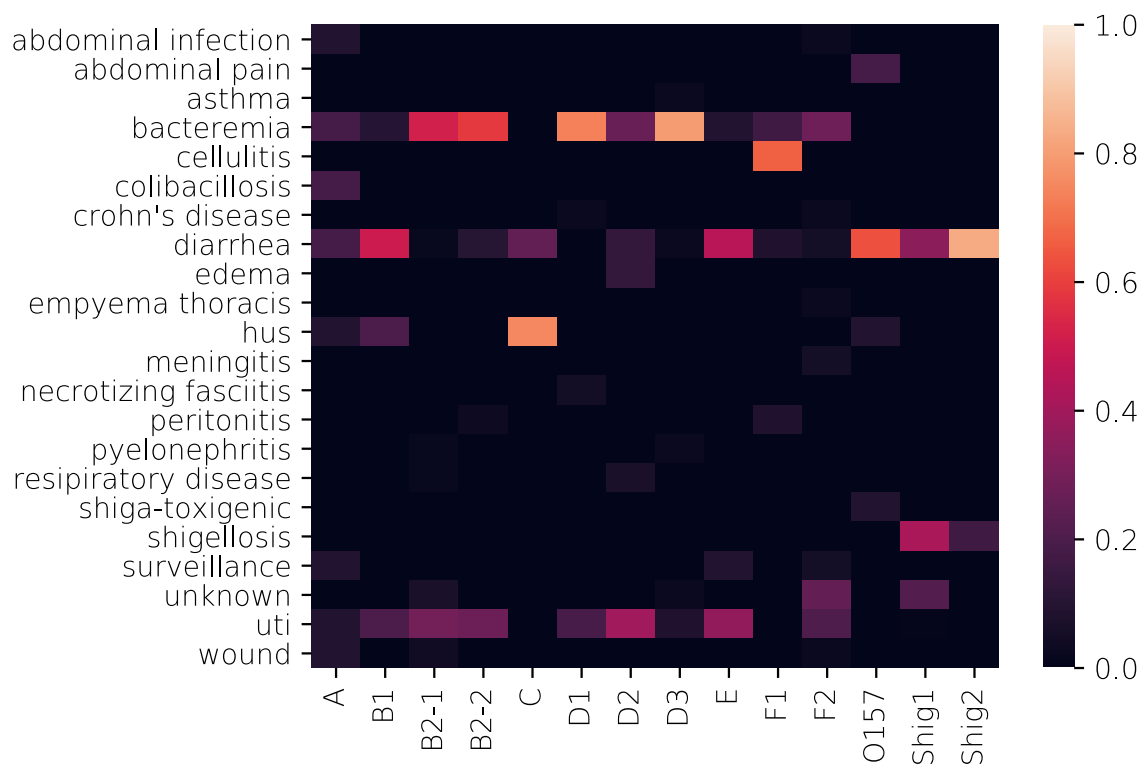


Figure 4.10: Sample host disease per phylotype (column normalised).

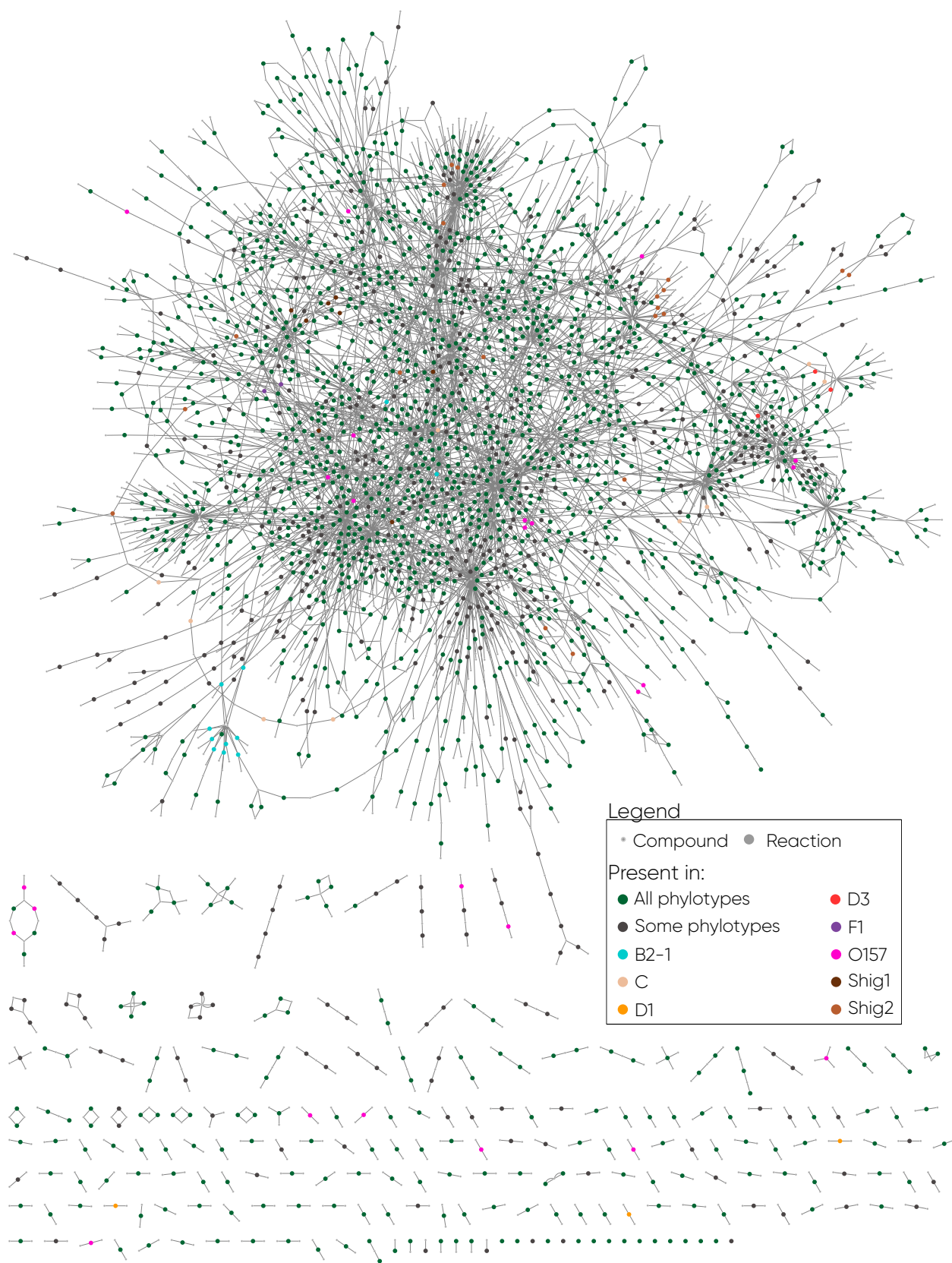


Figure 4.11: Reactions associated with each core genome.

Bibliography

- [1] Abram, K., Udaondo, Z., Bleker, C., Wanchai, V., Wassenaar, T. M., and Ussery, D. W. (2019). What can we learn from over 100,000 *Escherichia coli* genomes? *bioRxiv*, page 708131. [ix](#), [84](#), [86](#), [87](#), [88](#), [94](#)
- [2] Blount, Z. D. (2015). The natural history of model organisms: The unexhausted potential of *E. coli*. *Elife*, 4. [83](#)
- [3] Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T. L. (2009). Blast+: architecture and applications. *BMC Bioinformatics*, 10(1):421. [89](#)
- [4] Carlos, C., Pires, M. M., Stoppe, N. C., Hachich, E. M., Sato, M. I., Gomes, T. A., Amaral, L. A., and Ottoboni, L. M. (2010). *Escherichia coli* phylogenetic group determination and its application in the identification of the major animal source of fecal contamination. *BMC Microbiology*, 10(1):161. [84](#)
- [5] Caspi, R., Billington, R., Ferrer, L., Foerster, H., Fulcher, C. A., Keseler, I. M., Kothari, A., Krummenacker, M., Latendresse, M., Mueller, L. A., Ong, Q., Paley, S., Subhraveti, P., Weaver, D. S., and Karp, P. D. (2016). The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research*, 44(D1):D471–D480. [85](#)
- [6] Chaudhuri, R. R. and Henderson, I. R. (2012). The evolution of the *Escherichia coli* phylogeny. *Infection, Genetics and Evolution*, 12(2):214–226. [84](#)
- [7] Clermont, O., Christenson, J. K., Denamur, E., and Gordon, D. M. (2013). The Clermont *Escherichia coli* phylo-typing method revisited: Improvement of specificity and detection of new phylo-groups. *Environmental Microbiology Reports*, 5(1):58–65. [ix](#), [84](#), [86](#)
- [8] Clermont, O., Gordon, D., and Denamur, E. (2015). Guide to the various phylogenetic classification schemes for *escherichia coli* and the correspondence among schemes. *Microbiology*, 161(5):980–988. [83](#)
- [9] Gordienko, E. N., Kazanov, M. D., and Gelfand, M. S. (2013). Evolution of pan-genomes of *escherichia coli*, *shigella* spp., and *salmonella enterica*. *Journal of Bacteriology*, 195(12):2786–2792. [89](#)

- [10] Huerta-Cepas, J., Szklarczyk, D., Forslund, K., Cook, H., Heller, D., Walter, M. C., Rattei, T., Mende, D. R., Sunagawa, S., Kuhn, M., et al. (2015). eggnog 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. *Nucleic acids research*, 44(D1):D286–D293. [87](#), [93](#)
- [11] Ishii, S., Meyer, K. P., and Sadowsky, M. J. (2007). Relationship between phylogenetic groups, genotypic clusters, and virulence gene profiles of escherichia coli strains from diverse human and animal sources. *Appl. Environ. Microbiol.*, 73(18):5703–5710. [83](#)
- [12] Kaas, R. S., Friis, C., Ussery, D. W., and Aarestrup, F. M. (2012). Estimating variation within the genes and inferring the phylogeny of 186 sequenced diverse *Escherichia coli* genomes. *BMC Genomics*, 13(1):577. [89](#), [94](#)
- [13] Kaper, J. B., Nataro, J. P., and Mobley, H. L. (2004). Pathogenic escherichia coli. *Nature reviews microbiology*, 2(2):123. [83](#)
- [14] Karp, P. D., Billington, R., Caspi, R., Fulcher, C. A., Latendresse, M., Kothari, A., Keseler, I. M., Krummenacker, M., Midford, P. E., Ong, Q., Ong, W. K., Paley, S. M., and Subhraveti, P. (2017). The biocyc collection of microbial genomes and metabolic pathways. *Briefings in Bioinformatics*. [85](#)
- [15] Kim, Y., Wang, X., Zhang, X.-S., Grigoriu, S., Page, R., Peti, W., and Wood, T. K. (2010). Escherichia coli toxin/antitoxin pair MqsR/MqsA regulate toxin CspD. *Environ Microbiol*, 12(5):1105–1121. [97](#), [100](#)
- [16] Li, L., Stoeckert, C. J., and Roos, D. S. (2003). OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome research*, 13(9):2178–2189. [85](#), [89](#)
- [17] Montgomerie, J., Bindereif, A., Neilands, J., Kalmanson, G., and Guze, L. (1984). Association of hydroxamate siderophore (aerobactin) with escherichia coli isolated from patients with bacteremia. *Infection and immunity*, 46(3):835–838. [100](#)
- [18] Morcatti Coura, F., Diniz, S. d. A., Silva, M. X., Mussi, J. M. S., Barbosa, S. M., Lage, A. P., and Heinemann, M. B. (2015). Phylogenetic group determination of escherichia coli isolated from animals samples. *The Scientific World Journal*, 2015. [84](#)

- [19] Neo4 DevelopersJ (2012). Neo4J. *Graph NoSQL Database [online]*. [93](#)
- [20] Ochman, H. and Selander, R. K. (1984). Standard reference strains of *Escherichia coli* from natural populations. *Journal of bacteriology*, 157(2):690–3. [84](#)
- [21] Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., and Phillippy, A. M. (2016). Mash: fast genome and metagenome distance estimation using minhash. *Genome biology*, 17(1):132. [84](#)
- [22] Ørskov, F., Ørskov, I., Evans, D. J., Sack, R. B., Sack, D. A., and Wadström, T. (1976). Special *Escherichia coli* serotypes among enterotoxigenic strains from diarrhoea in adults and children. *Medical microbiology and immunology*, 162(2):73–80. [84](#)
- [23] Page, A. J., Cummins, C. A., Hunt, M., Wong, V. K., Reuter, S., Holden, M. T., Fookes, M., Falush, D., Keane, J. A., and Parkhill, J. (2015). Roary: rapid large-scale prokaryote pan genome analysis. *Bioinformatics*, 31(22):3691–3693. [89](#)
- [24] Perez, N., Spizzirri, F., Rahman, R., Suarez, A., Larrubia, C., and Lasarte, P. (1998). Steroids in the hemolytic uremic syndrome. *Pediatric Nephrology*, 12(2):101–104. [100](#)
- [25] Rasko, D. A., Rosovitz, M. J., Myers, G. S. A., Mongodin, E. F., Fricke, W. F., Gajer, P., Crabtree, J., Sebaihia, M., Thomson, N. R., Chaudhuri, R., Henderson, I. R., Sperandio, V., and Ravel, J. (2008). The pangenome structure of *Escherichia coli*: Comparative genomic analysis of *E. coli* commensal and pathogenic isolates. *Journal of Bacteriology*, 190(20):6881–6893. [94](#)
- [26] Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B. (2011). Finding a “kneedle” in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171. [91](#)
- [27] Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14):2068–2069. [85](#), [89](#)
- [28] Tenaillon, O., Skurnik, D., Picard, B., and Denamur, E. (2010). The population genetics of commensal *Escherichia coli*. *Nature Reviews Microbiology*, 8(3):207–217. [84](#)
- [29] Van Dongen, S. M. (2000). *Graph clustering by flow simulation*. PhD thesis. [89](#)

- [30] Vieira, G., Sabarly, V., Bourguignon, P.-Y., Durot, M., Le Fèvre, F., Mornico, D., Vallenet, D., Bouvet, O., Denamur, E., Schachter, V., and Médigue, C. (2011). Core and panmetabolism in *Escherichia coli*. *Journal of bacteriology*, 193(6):1461–72. [85](#)
- [31] Wanchai, V., Patumcharoenpol, P., Nookaew, I., and Ussery, D. (2017). dBBQs: database of bacterial quality scores. *BMC Bioinformatics*, 18(14):483. [88](#)
- [32] Wang, L., Wu, J., Liu, Z., Shi, Y., Liu, J., Xu, X., Hao, S., Mu, P., Deng, F., and Deng, Y. (2019). Aflatoxin B 1 degradation and detoxification by *Escherichia coli* CG1061 isolated from chicken cecum. *Frontiers in Pharmacology*, 9(JAN). [100](#)

Appendices 4

4.A Biocyc relabelling procedure

The metabolic models from BioCyc are based on the common background of MetaCyc. Unfortunately, it was found that some of the reaction IDs between MetaCyc and the individual genome models are inconsistent. This was confirmed with correspondence with BioCyc support. It was therefore necessary to create our own reaction database, and the below algorithm outlines how we went about it.

Algorithm 4.2 Biocyc relabelling procedure

```
1: procedure PROCEDURE(GenomeReactions, NewDB, MetacycDB)
2:   for reaction in GenomeReactions do
3:     if reaction.ID in NewDB then
4:       continue
5:     else if reaction.ID in MetacycDB then
6:       Add reaction to NewDB
7:     else if reaction.EQUATION matches a reaction in NewDB then
8:       Rename reaction in GenomeReactions to match NewDB reaction.ID
9:     else if reaction.EQUATION matches a reaction in MetaCycDB then
10:      Add reaction to NewDB
11:      Rename reaction in GenomeReactions to match NewDB reaction.ID
12:     else
13:       Add reaction to NewDB
14:     end if
15:   end for
16:   return NewDB
17: end procedure
```

4.B Currency compounds

Table 4.B.1: List of currency compounds

ID	Name	ID	Name
C00001	H2O	C00044	GTP
C00002	ATP	C00080	Hydrogen
C00003	NAD	C00286	dGTP
C00004	NADH	C01352	FADH2
C00005	NADPH	C00305	Magnesium catio
C00006	NADP	C00023	Iron
C00007	O2	C05359	Electron
C00008	ADP	C00070	Copper
C00009	Phosphate	C18714	Cuprous oxide
C00010	CoA	C00050	Metal
C00011	CO2	C00917	Chelating agent
C00013	Ppi	C00282	Hydrogen
C00014	NH3	C00390	UQH2
C00016	FAD	C00399	UQ
C00018	Pyridoxal phosphate	C00059	Sulfuric acid
C00020	AMP	C00087	Elemental sulfur
C00027	H2O2	C00094	Sulfurous acid
C00035	GDP		

4.C Sample host disease per phylotype

Table 4.C.1: *E. coli* genome sample associated host diseases.

Phylotype	Disease name	Number cases
A	colibacillosis	2
A	wounded	1
A	hus	1
A	intra-abdominal infections	1
A	screening	1
A	etec infection	1
A	bacteremia	1
A	urine track infection	1
A	sepsis	1
A	diarrhea	1
B1	secretory diarrhea	2
B1	diarrhoea	1
B1	bloody diarrhea	1
B1	asymptomatic bacteriuria	1
B1	sepsis	1
B1	urinary tract infection	1
B1	bloody diarrhea complicated hus	1
B1	dysentery	1
B1	hemolytic uremic syndrome	1
B2-1	bacteremia	18
B2-1	urinary tract infection	6
B2-1	wound	2
B2-1	urinary infection	2
B2-1	bloodstream infection	2
B2-1	sepsis	2

– Continued on next page

Table 4.C.1 – continued from previous page

Phylotype	Disease name	Number cases
B2-1	uti	1
B2-1	pyelonephritis	1
B2-1	bacteriuria	1
B2-1	infection	1
B2-1	d014552	1
B2-1	bacteraemia	1
B2-1	acute diarrhea	1
B2-1	urinary-tract infections (utis)	1
B2-1	pneumonia	1
B2-1	uropathogenic escherichia coli	1
B2-1	e. coli as the cause of diseases classified elsewhere	1
B2-1	asymptomatic	1
B2-2	sepsis	13
B2-2	urinary tract infection	8
B2-2	bloodstream infection	4
B2-2	diarrhea	2
B2-2	peritonitis	1
B2-2	dysentery	1
C	hus	8
C	bloody diarrhea	1
C	bloody diarrhoea	1
C	diarrhea	1
C	haemolytic uraemic syndrome	1
D1	bloodstream infection	17
D1	bacteremia	12
D1	urinary tract infection	8
D1	necrotizing fasciitis	2

– Continued on next page

Table 4.C.1 – continued from previous page

Phylotype	Disease name	Number cases
D1	sepsis	2
D1	crohn's disease	1
D2	urinary tract infection	5
D2	bloodstream infection	3
D2	diarrhoea	2
D2	edema disease	1
D2	edema disease of swine	1
D2	respiratory disease	1
D2	na	1
D2	recurrent uti	1
D2	septicemia	1
D3	bloodstream infection	11
D3	bacteremia	8
D3	blood stream infection	7
D3	bacteraemia	1
D3	watery diarrhea	1
D3	infection	1
D3	urinary tract infection	1
D3	recurrent uti	1
D3	septicemia	1
D3	asthma	1
D3	pyelonephritis	1
D3	urine infection	1
E	urinary tract infection	4
E	diarrhoea	2
E	healthy isolation source	1
E	bloody diarrhea	1

– Continued on next page

Table 4.C.1 – continued from previous page

Phylotype	Disease name	Number cases
E	gastroenteritis	1
E	diarrhea	1
E	bloodstream infection	1
F1	cellulitis	8
F1	bloodstream infection	1
F1	peritonitis	1
F1	diarrhea	1
F1	bacteremia	1
F2	infectious disease	8
F2	bloodstream infection	7
F2	urinary tract infection	4
F2	bacteremia	3
F2	diarrheal disease	2
F2	uti	2
F2	surveillance	2
F2	asymptomatic uti	1
F2	bacteriuria	1
F2	meningitis	1
F2	crohn's disease	1
F2	femur fracture	1
F2	sepsis	1
F2	colonization	1
F2	empyema thoracis	1
F2	neonatal meningitis	1
F2	abdominal infection	1
F2	asymptomatic	1
O157	diarrhea	3

– Continued on next page

Table 4.C.1 – continued from previous page

Phylotype	Disease name	Number cases
O157	abdominal pain	2
O157	eschirichia coli gastroenteritis	1
O157	gastroenteritis	1
O157	enterohemorrhagic escherichia coli	1
O157	bloody diarrhoea	1
O157	shiga-toxigenic escherichia coli	1
O157	hemorrhagic colitis, hus	1
Shig1	dysentery, bacillary	40
Shig1	diarrhea	33
Shig1	infectious disease	22
Shig1	diarrheal disease	3
Shig1	shigellosis	3
Shig1	urethral discharge	1
Shig2	diarrhea	4
Shig2	diarrhoea	1
Shig2	shigellosis	1

Chapter 5

Overall Conclusions and Directions for Future Work

This dissertation dealt with the fundamental biological data analysis problem of graph thresholding, and the application of data science principles to two diverse biological datasets. The approach taken throughout was to develop data-driven methods that are non-arbitrary and to a large extent, dataset independent. Together, this work demonstrates how algorithmic development can be applied to discover the implicit human understanding researchers employ and then apply it beyond the scope of current knowledge.

Each chapter dealt with justification, methodology, and conclusions. Here we review these contributions and emphasize future work.

Chapter 2 dealt with thresholding of complete biologically derived graphs. This is a fundamental step in producing graphs from biological datasets. We were able to compare a number of previously published methods, propose four new methods, implement a total of 15 methods in a united software package, and final compare their relative suitability on a number of transcriptomic and methylation graphs.

A number of algorithmic and implementation improvements can be made to the software, including parallelization, addition of permutation based distribution estimation, and considering positive and negative values separately (since many data distributions are not symmetric). These methods can also be applied to graphs from non-biological origin. For more practical use, this software can also be wrapped into R and Python libraries, and incorporated into user friendly environments such as Grappa [2] and KBase [1].

In Chapter 3, we applied graph algorithms to brain connectivity graphs. These methods were developed with the goal of establishing the degree of sample concordance and identifying factors that differentiate brainwaves. Already we could identify possible outliers, and bandwave specific features. Applying these methods to a dataset of differential diagnoses will help in understanding how brain connectivity dynamics differ between patient subtypes.

In Chapter 4, we developed a novel work flow to calculate bacterial core proteins. This core proteome pipeline allowed us to identify similarities and differences across previously defined *E.coli* phylotypes in a novel way, and may have implications for general core genome calculations.

The specific core proteome filtering method described in Section 4.2.2 can easily be expanded to other bacterial species and metagenomic datasets. Future work includes developing a generalized tool for such applications. In addition, this work used a subset of available data on

E. coli, and including more meta data into phylotype analysis may lead to stronger conclusions on phylotype and host disease associations. Finally, the connection between protein domains and proteins, and the graphs of protein family to function that this work produced, also leads to a further study on the relationship between protein domains and protein function across bacteria (currently within *E. coli*).

Bibliography

- [1] Arkin, A. P., Cottingham, R. W., Henry, C. S., Harris, N. L., Stevens, R. L., Maslov, S., Dehal, P., Ware, D., Perez, F., Canon, S., et al. (2018). Kbase: the united states department of energy systems biology knowledgebase. *Nature biotechnology*, 36(7):566. [119](#)
- [2] Lu, Y., Phillips, C., and Langston, M. A. (2015). GrAPPA. <https://grappa.eecs.utk.edu>. [119](#)

Vita

Carissa was born in Johannesburg and raised in Cape Town, South Africa. She completed her BSc and BScHons in Mathematics at Stellenbosch University in 2013 and 2014 respectively. In 2015 she started her doctoral studies at the Bredesen Center for Interdisciplinary Research and Graduate Education, at the University of Tennessee. She worked as a graduate research assistant, initially with Dr Daniel Jacobson in the Computational Biology and Bioinformatics group at ORNL and then in Dr Michael Langston's lab in the EECS department. In both groups she worked on the application of graphs and graph theory to biological datasets. Carissa completed her doctoral degree in data science and engineering in May 2020.