Distribution Agreement

In presenting this dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University and its agents the non-exclusive license to archive, make accessible, and display my dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this dissertation. I retain all ownership rights to the copyright of the dissertation. I also retain the right to use in future works (such as article or books) all or part of this dissertation.

Signature:

_____ _____

Margaret Justice Bray Date

Algorithmic Approaches to Classifying Biological Networks

By

Margaret Justice Bray

Doctor of Philosophy

Biostatistics

Vicki Hertzberg, Ph.D.

Advisor

John Hanfelt, Ph.D.

Committee Member

William McClellan, Ph.D.

Committee Member

Tianwei Yu, Ph.D.

Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.

Dean of the James T. Laney School of Graduate Studies

Date

Algorithmic Approaches to Classifying Biological Networks

By

Margaret Justice Bray

B.S., Rensselaer Polytechnic Institute, 2010

M.S., Rensselaer Polytechnic Institute, 2010

M.S., Emory University, 2014

Advisor: Vicki Hertzberg, Ph.D.

An abstract of

A dissertation submitted to the Faculty of the James T. Laney School of Graduate Studies of Emory University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Biostatistics

2015

Abstract

Algorithmic Approaches to Classifying Biological Networks

By

Margaret Justice Bray

As technology has become more advanced, the ease with which data can be collected has improved. This has left researchers with copious amounts of information, so much information that previous analytical techniques fall short. This has led to an increase in the popularity of representing data with networks. However, this data often have errors. This makes any conclusion gleamed from the analysis of a network unreliable.

One type of network for which the inaccuracies are a particular issue is the protein-protein interaction (PPI) network. Researchers would like to use these networks to detect and diagnosis diseases by identifying specific interactions. Unfortunately, the errors in the networks make this impossible. One way to fix this is classify the empirical network into a category of model graph. By doing so, we will be able to mathematically predict which interactions are legitimate, and which are not.

In this dissertation, we begin by testing the classification accuracy of five algorithms: degree distribution distance (DDD), characteristic curve (CC), relative graphlet frequency (RGF), graphlet degree distribution using arithmetic mean (GDD (A)) and using geometric mean (GDD (G)). Overall accuracies were poor, ranging from 68% for the GDD (A) down to 47% for the DDD. With accuracies this low, it is difficult to trust the classification results for an empirical network of unknown origin.

Therefore, we propose two solutions. First, we provide several modifications to both versions of the GDD. The reformulated GDD is more accurate, classifying 76% of known graphs correctly, while also performing the analysis with increased speed. Second, we present a new classification algorithm: cross scoring. This novel method works by comparing networks based on a pre-selected group of network measures. Each type of model graph is ranked by how close its measure value falls to the empirical value compared to the other model types considered. Points are awarded and the model type with the fewest points at the end of the comparisons is considered the best fit. Accuracy across twelve trials was 82.9% (± 0.98). These results are an obvious improvement over the five original algorithms considered.

Algorithmic Approaches to Classifying Biological Networks

By

Margaret Justice Bray

B.S., Rensselaer Polytechnic Institute, 2010
M.S., Rensselaer Polytechnic Institute, 2010
M.S., Emory University, 2014

Advisor: Vicki Hertzberg, Ph.D.

A dissertation submitted to the Faculty of the James T. Laney School of Graduate Studies of Emory University in partial fulfillment of the requirement for the degree of Doctor of Philosophy in Biostatistics.

2015

Acknowledgments

I would like to begin by thanking my PhD advisor, Dr. Vicki Hertzberg, for guiding and mentoring me for the past several years. I would not have been able to complete this dissertation, or the research it describes, without her continuous support. In addition, I would like to thank my committee members, Dr. John Hanfelt, Dr. Tianwei Yu, and Dr. William McClellan. Dr. Hanfelt and Dr. Yu were patient, critical readers throughout the many drafts of my dissertation. A special thank you to Dr. Frank Emmert-Streib for participating on my committee through the initial stages and for giving me the push needed to take my dissertation from merely adequate and turning into a document that I can truly be proud of. An additional special thank you to Dr. McClellan for being willing to participate on my final defense committee at short notice.

In addition to the members of my committee, I would like to thank the numerous other professors that supported me throughout my time at Emory University, including Department Chair Dr. Lance Waller and my academic advisor, Robert Lyles. Thank you for giving me the strength and confidence to complete this program, especially during the first three difficult years.

Finally, I would like to thank my family, especially my parents, for always being there through the many ups and downs for the past five years. Without their never wavering faith that I would finish, I am not sure that I would have.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction to Graphs and Graph Theory

With the advent of big data, researchers have been inundated with information. In an effort to find ways to analyze this data, graph theory and, more specifically, the analysis of real-world networks, has become a popular and ever-growing field. Examples of real-world networks whose analysis has been tantalizing to researchers include the Internet (Faloutsos *et al.* , 1999), the World-Wide Web (Kumar *et al.* , 1999; Broder *et al.* , 2000), scientist citation networks (Seglen, 1992; Newman, 2001a), as well as various biological and metabolic networks (Jeong *et al.* , 2001), The main reason for the study of the structure of these real, complex networks is that structure always affects function (Strogatz, 2001). Therefore, if the structure of a network is known, then underlying functions that may previously have gone unnoticed may be revealed.

The research in this dissertation is motivated by biological networks defined by protein-protein interactions (PPI). Proteins are essential to cells. They perform a huge number of functions within every living thing including acting as catalysts, messengers, and cellular structure. Protein-protein interactions (PPIs) are essential in the orchestration of such events (Raman, 2010). Due to advances in biotechnology, the accumulation of data involving PPIs has never been easier (Kuchaiev & Przulj, 2009). Unfortunately, these new technologies report numerous false-postives, i.e. identification of interactions that do not actually occur *in vivo*, which make it difficult to truly assess protein function.

It is essential that we identify the underlying structure of PPI networks for a number of reasons. By determining the structure, it will be possible to predict interactions that were not previously identified. This can give biological researchers areas in which to focus their efforts, when modifying their current methods for determining PPI's, or creating new methods. In addition, the discovery of orthologous proteins (i.e., proteins that have the same or similar functions across evolutionary related species) in simple organisms can ease the deciphering of the PPI networks for more complex organisms. Ideally, we would like to

be able to analyze the human PPI network with the goal of identifying interactions whose presence or absence is associated with a certain disease.

This idea leads to the motivation for the research presented here. In this dissertation, we delve into the problem of PPI network classification. Network classification is the process by which a type of *model graph* (Chapter 3) is determined to best mimic the characteristics of the real-world network under investigation (Vogelstein *et al.* , 2013). The chapters can be split up into three groups. In the first four chapter, all of the necessary terms and ideas are introduced. We gain an understanding of the PPI network being analyzed, as well as the types model graphs used for classification, and how the two relate to each other. The remainder of Chapter 1, provides a review of terms from graph theory that are necessary to understand the remainder of the work. This is followed by an introduction to the specific PPI network that is the focus of this research in Chapter 2 and the selection of model graphs considered for classification categories (Chapter 3). In Chapter 4, we move onto to a numerical comparison of the real-world network to the model graphs based upon graph features, or *measures*, presented in Chapter 1.

The next four chapters introduce and evaluate a selection of classification methods. These classifiers are officially introduced in Chapter 5. Then, their ability to perform well and accurately classify networks in tested on both random graphs (Chapter 6) and model graphs (Chapter 7). The methods are all then applied to the PPI network under investigation and the results analyzed for reliability (Chapter 8).

The final five chapters examine improvements that can be made to the entire classification process. Chapters 9 and 10 show corrections and improvements made to two of the classifiers. These changes resulted in increased accuracy without detracting from the original idea of the algorithm. Finally, we introduce a novel network classification method. We describe how it was designed in Chapters 11 and 12, then apply it in Chapter 13. We conclude by comparing the novel classifier to both the updated and original classifiers and a present a selection of future work (Chapter 14).

1.1 Review of Essential Graph Theory Elements

Any group of entities that have some sort of connection relationship with one another can be described by use of a graph. A formal working definition of this concept, as expressed by Ernesto Estrada, refers to a network as a "diagrammatic representation of a system" (Definition 1.1).

> **Definition 1.1.** *"A network (graph) is a diagrammatic representation of a system. It consists of node (vertices), which represent the entities of a system. Pairs of nodes are joined by links (edges), which represent a particular kind of interconnection between those entities." (Estrada, 2011)*

In many instances, such as in Definition 1.1, the terms network and graph are used interchangeably, along with node and vertex, edge and link. Some groups of researchers, however, prefer to use the term network to refer to any representation of a real-world system and graph to refer to any mathematically based model (Winer, 2007). In this dissertation we will use the latter interpretation. Note that network and graph can still be used interchangeably when discussing generic features or properties such as in Section 1.2. Node and vertex along with edge and link will still be used interchangeably throughout the dissertation.

The set of all nodes, or vertices, in a network can be written as $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, where n is the total number of nodes in the system. The set $\mathcal{V} \otimes \mathcal{V}$ represents all of the ordered pairs, (v_i, v_j), in $\mathcal{V}$, where v_i and v_j are not necessarily unique. The subset $\mathcal{E} \subseteq \mathcal{V} \otimes \mathcal{V}$ represents the set of all edges in the system. This relation, $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$, contains m elements. In addition, $\mathcal{E}$ is symmetric if $(v_i, v_j) \in \mathcal{E}$ implies that $(v_j, v_i) \in \mathcal{E}$. This is an example of an *undirected network*. If $(v_i, v_j) \in \mathcal{E}$ does not imply $(v_j, v_i) \in \mathcal{E}$, then the network is *directed*. It is *antireflexive* if $(v_i, v_j) \in \mathcal{E}$ implies that $v_i \neq v_j$ (Estrada, 2011). The total number of nodes or edges in a specific network can also be referenced by $|\mathcal{V}_\mathcal{G}|$ and $|\mathcal{E}_\mathcal{G}|$, respectively. Two distinct nodes may be represented by (u, v).

A *simple network*, by definition, is undirected and does not contain self-loops, i.e., an edge that begins and ends at the same node. The written description of such a network can be expressed by the use of the pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (Definition 1.2).

> **Definition 1.2.** *"A simple network is the pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of nodes and $\mathcal{E}$ is a symmetric and antireflexive relation on $\mathcal{V}$." (Estrada, 2011)*

In addition, in a simple network there is no pair of vertices, (u, v), with more than one edge between them. Therefore, $e_i \neq e_j$, $\forall i \neq j$. Nodes that do have multiple edges between them are said to have multi-edges. A network containing either self-loops or multi-edges is no longer considered simple; it is a *multi-graph* (Kolaczyk, 2009). In addition to a simple graph, we also have concepts of *null network* and *complete network*. A null network is a network that does not contain any edges. There are no connections between any nodes. A complete network is a graph in which every node is connected to every other node.

The concept of *adjacency* is another important concept in the realm of graph theory. Two nodes, u and v, are considered adjacent if there is an edge, $e \in \mathcal{E}$, between u and v. The set of all of the nodes adjacent to node u is known as the set of neighbors of u, $N(u)$. Two edges are considered adjacent if they have a common endpoint. One way to represent a network is through an adjacency matrix. In this matrix, each row, and column, represents a node. If two nodes are *adjacent*, i.e. have an edge between them, then the matrix has a one in the corresponding location. If there is no edge between two nodes, then the appropriate cell in the matrix has a zero. In some instances, we may see a value other than a zero or a one in the adjacency matrix. This value indicates that the edge is weighted. A *weighted network* is one that acknowledges that edges have different strengths (Newman, 2004). Weights can be derived in numerous ways. They can be derived simply and correspond to the number of edges between two nodes in a multi-graph or be more complex. Complex weighting schemes based on network measures discussed in Section 1.2 such as closeness centrality (Newman, 2001b), betweenness centrality (Brandes, 2008), global clustering coefficient (Opsahl & Panzarasa, 2009), and local clustering coefficient (Barrat *et al.* , 2004; Zhang & Horvath, 2005), have also been proposed.

In simple matrices the diagonal of the adjacent matrix is always composed of zeros and the only other number seen in the matrix is one. If the network is undirected, then the matrix is symmetric. An example of this can be seen below:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

In the above matrix cell $(1, 2)$ contains a one. This means that there is an edge between nodes v_1 and v_2. The matrix is symmetric, thus the corresponding network is undirected. Finally, the diagonal is composed of zeros and no other value besides one is present. This implies that this is the adjacency matrix for a simple network.

A network is *connected* if it is possible to touch every vertex by traversing the set of edges. If a network is not connected, then each connected piece is referred to as a *component*. The largest connected component, judged by number of nodes, is called the *giant connected component*, or *giant component*, of the network. While the full network is referred to as $\mathcal{G}$, the giant component is referred to as $\mathcal{H}$. The adjacency matrix can be a useful tool in determining graph components (Fiedler, 1973).

1.2 Network Measures

Network measures are features of a graph or network that can be used for classification, characterization, and categorization. This term can be used interchangeably with *graph measures*. These measures can either be reference properties of the full network or properties of individual nodes. These referred to as network-level (graph-level) or node-level (nodal) properties respectively. We examine nineteen measures in this chapter. Eleven of them are network-level and the remaining eight are nodal properties. Averages of node-level measures can be taken to obtain a single, summary value to describe the entire network.

Two of the simplest graph-level measures have already been introduced. These measures are number of nodes and number of edges. These measures are, obviously, the basis for all of the other measures, however there are many ways that the edges can be distributed among the nodes. Thus, these features alone do not tell the whole story of the network.

The *degree*, k, of node v is equal to "the number of edges incident on v", where the concept of incidence refers to the number of edges of which v is an endpoint (Kolaczyk, 2009). Degree is usually expressed as the average degree of a network, Equation 1.1. It is the the sum of all node degrees divided by the total number of nodes:

$$\bar{k} = \frac{1}{n} \sum_{i=1}^{n} k_i. \tag{1.1}$$

In addition to average degree, the average neighbor degree can be calculated. The average neighbor degree of node v is defined as:

$$k_n(v) = \frac{1}{|N(v)|} \sum_{u \in N(v)} k_u, \tag{1.2}$$

where $|N(v)|$ is the number of neighbors of node v and k_u is the degree of node u.

Network *density* (Equation 1.3) measures how close a network is to complete (Kolaczyk, 2009). It is calculated by the number of edges, m, divided by the total number of possible edges:

$$\begin{aligned} \mathcal{D}(\mathcal{G}) &= \frac{m}{n(n-1)/2} \\ &= \frac{2m}{n(n-1)}. \end{aligned} \tag{1.3}$$

The range of density falls between zero and one, with the lower limit corresponding to a *null network*, a network with no edges, and the upper corresponding to a *complete network*, a network where every node is connected to every other. Most real world networks have low densities that are much closer to zero than to one (Melancon, 2006).

The proportion of nodes in the giant component is related to density. Graphs with higher densities will tend to have fewer distinct components, thus more nodes will be in the giant component. This is an important concept in the analysis of networks because it provides an easily interpretable description of the network shape with just one number:

$$|\mathcal{V}_{\mathcal{H}}| \,/\, |\mathcal{V}_{\mathcal{G}}| \,. \tag{1.4}$$

The *clustering coefficient* of any vertex, Equation 1.5, uses the concept of a *transitive relation*. Such a relation says the if v_1 is connected to v_2 and v_1 is connected to v_3, then v_2 is connected to v_3 (Leinhardt, 1976). Such a relation clearly represents a triangle. Then the clustering coefficient of any node, v, is the number of transitive relations, or number of triangles, t, that the node, v, takes part in, divided by the total number of triads it takes part in (Estrada, 2011):

$$\begin{aligned} C_v &= \frac{t_v}{k_v(k_v-1)/2} \\ &= \frac{2t_v}{k_v(k_v-1)}. \end{aligned} \tag{1.5}$$

A triad, or triple, is a possible transitive relation. This means that it is a path of length two where the end nodes are not connected to each other. The lack of connection between these end nodes is what leads the *triad* to be a possible transitive relation instead of simply a transitive relation. In Equation 1.5, t_v is the number of triangles that node v takes part in and k_v is the degree of that same node.

The average clustering coefficient is achieved by taking the average clustering coefficient over all of the nodes in $\mathcal{G}$:

$$\bar{C} = \frac{1}{n}\sum_{v\in\mathcal{V}} C_v. \tag{1.6}$$

Clustering can also be looked at as a global property of a network, as opposed to a property of individual nodes. This measure is called *transitivity*, or the global clustering coefficient:

$$C(\mathcal{G}) = \frac{3\,|C_3|}{|P_2|}. \tag{1.7}$$

In Equation 1.7, $|C_3|$ is the number of cycles of size three and $|P_2|$ is the number of paths of length two. This can also be looked at as the number of triangles in the network divided by the number of triads (Kolaczyk, 2009; Estrada, 2011). It is very similar to the clustering coefficient of individual nodes seen in Equation 1.5.

The *eccentricity*, Equation 1.8 of any given vertex, v, is the longest shortest path in which it serves as an endpoint.

$$e(v) = \max_{x \in \mathcal{V}_\mathcal{G}} \{d(v, x)\} \tag{1.8}$$

The *diameter* of a network is the maximum eccentricity across all of the nodes.

$$diam(\mathcal{G}) = \max_{x,y \in \mathcal{V}_\mathcal{G}} \{d(x, y)\} \tag{1.9}$$

Nodes with eccentricities equal to the diameter are *peripheral* nodes. The *radius* of a network is the minimum eccentricity across all nodes.

$$rad(\mathcal{G}) = \min_{x,y \in \mathcal{V}_\mathcal{G}} \{d(x, y)\} \tag{1.10}$$

If a node happens to have eccentricity equal to the radius, then it is referred to as *central*. The set of central nodes are collectively referred to as the center of the graph (Estrada, 2011). The previous two properties, diameter and radius, require a connected network in order to be calculated.

Related to the concept of path length is the *average shortest path length* (ASPL), also sometimes referred to as the characteristic path length. It is a global metric that measures the average distance between any two vertices in a graph (Watts & Strogatz, 1998):

$$\bar{\ell}(\mathcal{G}) = \frac{1}{n(n-1)} \cdot \sum_{i \neq j} d(v_i, v_j). \tag{1.11}$$

The average shortest path length is bounded by:

$$1 \leq \bar{\ell}(\mathcal{G}) \leq \frac{n+1}{3}, \tag{1.12}$$

where the lower bound is achieved by a complete network and the upper bound is achieved by a path of length n (Estrada, 2011).

1.2.1 Small-World and Scale-Free

In order to understand the next several graph measures, it is necessary to provide some background into types of model graphs including random, lattice, small-world, and scale-free. Recall that model graphs are mathematically based models. They do not directly represent any real-world system.

Random and lattice graphs are two simple graphs. They do not contain complex topologies. Random graphs are created by connected any node v_1 to any other node with probability p (Gilbert, 1959; Erdős & Rényi, 1960; Bollobás, 1998). Lattice graphs, also referred to as mesh graphs or grid graphs, are graphs embedded in a Euclidean space that form regular tilings. A lattice of size $n \times n$ is embedded in space R^n. In a cubic lattice, "vertices are the ordered triplets on n symbols, such that two vertices are adjacent if and only if they have two coordinates in common" (Figure 1.1) (Aigner, 1969; Laskar, 1969).

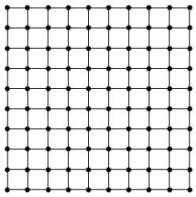


Figure 1.1. **Example of a cubic lattice.**

Small-world graphs, which will be discussed in further detail in Chapter 3.1, are categorized by two features (or properties), their short average shortest path length (Equation 1.11) and high global clustering coefficient (Equation 1.7). These features cannot be replicated by either random or lattice graphs (Watts & Strogatz, 1998). Scale-free graphs, also described Chapter 3.1, are considered ultra-small based on their average shortest path length (Cohen & Havlin, 2003). This is in large part due to the formation of hubs. Hubs are nodes that have many more connections than other nodes in the same graph (Barabási & Albert, 1999). Due to the existence of hubs, the graph's path length grows proportionally to the number of nodes, n (Cohen & Havlin, 2003).

To determine whether a network has small world features we must compare it to a random graph with size n nodes and average degree $\bar{k}$. Conveniently, we do not actually have

to create a random graph with the desired number of nodes and average degree in order to calculate the average shortest path length and transitivity needed for the comparison (Watts & Strogatz, 1998; Cohen & Havlin, 2003). Instead, we can approximate those parameters by:

$$\bar{\ell}_r = \frac{\log n}{\log \bar{k}}, \tag{1.13}$$

and:

$$\bar{C}_r = \frac{\bar{k}}{n}. \tag{1.14}$$

Then we can calculate two proportions:

$$p = \frac{\bar{C}}{\bar{C}_r} \tag{1.15}$$

$$q = \frac{\bar{\ell}}{\bar{\ell}_r}. \tag{1.16}$$

If $p \gg 1$ and $q \approx 1$ we can determine that the network has small-world properties.

In addition to the small-world property relating to the average shortest path length, there is also a scale-free property. We can check a network for the scale-free property similar to how we check for the small-world property, but this time there is no clustering requirement. Since the average shortest path length grows proportionally to the number of nodes, n, in the network such that:

$$\bar{\ell}(\mathcal{G}) \propto \log \log n, \tag{1.17}$$

we now estimate the average shortest path length of the random graph with size n by:

$$\bar{\ell}_{r,ultra} = \frac{\log n}{\log \log n}. \tag{1.18}$$

Then if $s \gg 1$ we can determine that the network has the scale-free property and is ultra-small, where s is defined as:

$$s = \frac{\bar{\ell}}{\bar{\ell}_{r,ultra}}. \tag{1.19}$$

This property was shown by Cohen and Havlin (Cohen & Havlin, 2003; Cohen *et al.* , 2003).

The next measure, *assortativity*, is not directly related to the SMW or SF property, but it is necessary to understand in order to comprehend another measure, the S-metric, which is directly related to the SF property. Assortativity, $r(\mathcal{G})$, is defined such that:

$$r(\mathcal{G}) = \frac{\sum_{xy} xy(e_{xy} - a_x b_y)}{\sigma_x \sigma_y}, \tag{1.20}$$

where e_{ij} is "the fraction of all edges in the network that join together vertices with values x and y" (Newman, 2003). In other words, it is looking at the portion of high-degree nodes connected to other high-degree nodes (Newman, 2002). Values are typically node degrees, but the assortativity formula can be applied to other graphical features. The expressions a_x and b_y represent "the fraction of edges that start and end at vertices with values x and y" (Newman, 2003), respectively. Formulas for these two quantities can be seen in Equation 1.21 and 1.22:

$$a_x = \sum_y e_{xy} \tag{1.21}$$

$$b_y = \sum_x e_{xy}. \tag{1.22}$$

The denominator of Equation 1.20 is made up of the standard deviations of the distributions for a_x and b_y.

The *s-Metric* "measures the extent to which the graph $\mathcal{G}$ has a "hub-like" core where a "hub-like" core is defined as a set of nodes that have more connections than other nodes in the graph and "play a central role in the overall connectivity of the network" (Li *et al.* , 2005). The s-metric is maximized when high-degree nodes are connected to other high-degree nodes" (Li *et al.* , 2005). In other words, the s-Metric is a measure of the assortativity

of a network and can "measure the extent to which a graph is scale-free" (Li *et al.* , 2005). It typically assumes an undirected, simple, connected graph and is calculated as follows: the ordered degree sequence of the nodes of a network is given by $\mathcal{K} = \{k_1, k_2, \ldots, k_n\}$ where k_i is the degree of node i and then:

$$s(\mathcal{G}) = \sum_{e_{i,j} \in \mathcal{E}} k_i k_j. \tag{1.23}$$

The s-metric can be normalized through use of the s_{max}-graph, which is the graph with ordered degree sequence $\mathcal{K}$ that has the largest s-Metric value. The value seen in Equation 1.23 is maximized, as previously mentioned, when high degree nodes are connected to one-another. It is important to note that the construction of an unconstrained s_{max}-graph is nontrivial as long as the network is required to be simple (Waldorp & Schmittmann, 2015).The normalized metric is calculated by:

$$S(\mathcal{G}) = \frac{s(\mathcal{G})}{s_{max}(\mathcal{G})}. \tag{1.24}$$

Normalizing the s-metric, so it becomes the S-metric, allows for the comparison of networks with different degree sequences (Beichl & Cloteaux, 2008). For that reason, the normalized metric is the one utilized in this research. There are numerous advantages of this metric over others. The main one is its ability to differentiate between networks with identical degree distributions, but different topological properties. Additionally, this metric requires more than a degree distribution that is scaling in order for it to be deemed a scale-free network, it also requires the network to be self-similar (Li *et al.* , 2005). Self-similarity in a network is expressed as self-repeating patterns (Song *et al.* , 2005). A network that exhibits a scaling degree distribution without self-similarity results in a low value for $S(\mathcal{G})$ and is referred to as scale-rich as opposed to scale-free.

Both the S-metric and assortativity measure the connectivity of high-degree nodes to other high-degree nodes and low-degree to low-degree. In assortativity, if high-degree nodes are connected to other high-degree nodes (and low-degree to low-degree), then the network is assortative. If high-degree nodes are not preferentially connected to other high

degree nodes, then the network is dissasortative. The range for assortativity is $[-1, 1]$, with the lower bound corresponding to dissasortative mixing and the upper bound to assortative mixing. If the assortativity coefficient is 0, then the network has neutral mixing (Estrada, 2011)

The association between $S(\mathcal{G})$ and $r(\mathcal{G})$, is direct since they usually measure essentially the same thing, however this is not always the case. There are numerous instances where the two conflict. The reasons for the disagreements are the "background sets" used for normalization (Li *et al.* , 2005). As previously mentioned, $S(\mathcal{G})$ is normalized against a simple, connected graph. The normalized assortativity value is not forced under the same constraints. In fact, in nearly all situations the network with the highest unnormalized assortativity or s-metric would have multiple self-loops and be connected. Thus, both metrics are useful despite their conceptual similarity.

1.2.2 Centrality

The determination of the most important nodes in a network is very interesting and highy subjective. A node considered important by one measure may not be considered so by another. Most measures agree that a node's importance is related to high degree, however that is typically where the agreement ends (Freeman, 1979). Measures that test the importance of any given node are called *centrality measures*. While there are numerous centrality measures, only four were considered for this analysis: *degree*, *betweenness*, *closeness*, and *eigenvector* (Kolaczyk, 2009; Estrada, 2011).

Due to the undirected nature of the networks under investigation, it was not necessary to separate in-degree from out-degree. Degree centrality of any given node, i, simply refers to the degree of that node and is normalized by dividing by the maximum possible degree (Estrada, 2011):

$$DC_i = \frac{k_i}{n-1}. \tag{1.25}$$

Closeness centrality examines the distance from one node to every other node in the network. Nodes that can reach most other nodes in the fewest number of steps are rewarded

with high closeness centrality values by taking the inverse of each shortest path distance from the node in question (Estrada, 2011). Mathematically this is represented by:

$$CC_i = \frac{n-1}{\sum_{v \in V} d(i,v)}, \tag{1.26}$$

where V is the set of all vertices in the network. The average path length is normalized by the maximum possible path length, $n-1$, which is where the numerator of Equation 1.26 is derived. Closeness can be calculated on unconnected networks by calculating the centrality of each connected component separately.

The betweenness centrality metric examines the relative importance of a node in communication between other nodes. It measures this by the fraction of shortest paths from s to t, where u acts as a bridge divided by the total number of shortest path from s to t:

$$BC_i = \sum_{s,t \in V} \frac{\rho(s,i,t)}{\rho(s,t}, s \neq t \neq i. \tag{1.27}$$

In Equation 1.27, $\rho(s,u,t)$ refers to the number of shortest paths from s to t that pass through i and $\rho(s,t)$ refers to the total number of shortest paths from s to t (Estrada, 2011). We then normalize this total number by $2/((n-1)(n-2))$. The normalization factor allows for more accurate comparisons between networks of drastically different sizes.

Eigenvector centrality was first proposed by Bonacich in 1987 (Bonacich, 1987). The idea for this metric was based on actor networks. In such networks, an actor's centrality indicated the extent to which a given actor was associate with other central actors (Estrada, 2011). Essentially, this measure looks for the most influential node in the network by acknowledging that not all connections are equal (Newman, 2008).

The eigenvector centrality is calculated, in this instance, by using the power method, also known as the power iteration eigenvalue algorithm, to identify the eigenvector associated with the largest eigenvalue of the adjacency matrix of network $\mathcal{G}$. This is the principal eigenvector. The ith entry from this principal eigenvector is the eigenvector centrality of node v_i, $\psi_i(i)$.

To begin, we define:

$$x_{v_i} = \frac{1}{\lambda} \sum_{j=1}^{n} A_{ij} x_{v_j}, \tag{1.28}$$

where A is the adjacency matrix of the network and A_{ij} is the entry in the i-th column and j-th row indicating whether node v_i and node v_j have an edge between them. The values x_{v_i} and x_{v_j} represent the centralities of nodes v_i and v_j. The value λ is a constant. If we define a vector of all the node centralities such that $\mathbf{x_v} = (x_{v_1}, x_{v_2}, \ldots, x_{v_n})$, then the equation rewritten in matrix form is:

$$\lambda \mathbf{x} = \mathbf{A} \cdot \mathbf{x}. \tag{1.29}$$

The equation in Equation 1.29 is traditional equation for the calculation of eigenvalues λ and eigenvectors $\mathbf{x}$. The eigenvalue is required to be positive, therefore it can be shown by the Perron-Frobenius theorem (Keener, 1993) that λ is the largest eigenvalue of the adjacency matrix and $\mathbf{x}$ is the corresponding eigenvector (Newman, 2008).

1.3 Discussion

In this chapter we presented a brief introduction to some essential elements from graph theory. We then moved onto the discussion of ways to summarize networks. These statistics take the form of network measures. Network measures can either represent graph-level properties or node-level properties. Often times averages are taken of the node-level properties to obtain a single statistics that summarizes the whole network. A total of nineteen measures were examined. Eleven of these are graph-level and eight are node-level. The graph-level measures are number of nodes, number of edges, density, proportion of nodes in the giant component, transitivity, diameter, radius, SMW property, SF property, assortativity, and S-metric. The node-level measures are degree, neighbor degree, clustering coefficient, shortest path length, and the four centrality measure: degree, betweenness, closeness, and eigenvector. In the next chapter, we present the real-world network that is at the center

of this dissertation. The network is then evaluated using the measures presented in this chapter.

Chapter 2

Introduction to the Structure of Biomolecular Networks

We have mentioned that the main topic of this dissertation concerns the classification of real-world networks. Nearly any system can be represented by a network. There are computer networks such as the internet and world wide web, social networks such as friendships, and biological networks. The latter category contains networks such as food webs, but also encompasses molecular networks. The focus of this dissertation is on molecular networks, proten-protein interaction networks in particular. In this chapter, we introduce the concept of a protein-protein interaction network. We then present the specific network used for the remainder of the analyses. Finally, we use the graph measure explained in Chapter 1 to attempt to summarize the network's features.

2.1 Protein-Protein Interactions

Protein-protein interactions (PPI) are defined as "physical contacts with molecular docking proteins that occur in a cell or in a living organism *in vivo*" (De Las Rivas & Fontanillo, 2010). A protein-protein interaction network is built using known protein-protein interactions. Nodes are used to represent proteins, and edges represent interactions. Since interactions are considered mutual (if $\mathcal{A}$ interacts with $\mathcal{B}$, then $\mathcal{B}$ interacts with $\mathcal{A}$) the network is undirected. De Las Rivas believed that interactions used to build the networks should meet two criteria. First, they should be intentional, "the result of specific selected bimolecular forces" (De Las Rivas & Fontanillo, 2010). Secondly, the interaction interface should be non-generic. In other words, it should have evolved for a specific purpose unique from generic functions such as protein production or degradation.

Currently there are multiple techniques for building protein-protein interaction maps. High-throughput methods, such as yeast two-hybrid screening, bimolecular fluorescence complementation (BFC), tandem-affinity purification (TAP) combined with mass spectrometry (MS), and chemical cross linking, give the most accurate results. The former two methods are *in vivo* methods while the latter two occur *in vitro*. Comparing the data resulting from the different methods is difficult since the the data were derived under dif-

ferent conditions with different goals. Of approximately 80,000 total interactions available from different high-throughput methods only $\sim 3\%$, or $\sim 2,400$, are supported by more than one method (Von Mering *et al.* , 2002). There are several possible reasons for this discrepancy. First, the methods may each detect a significant number of false-positives. Second, each method may have a predilection for reporting certain types of interactions or be disinclined to report others. Third, these methods may not have reached a point where they are detecting all interactions available, thus a large number of false negatives. For these reasons it is crucial to use as many methods as possible in order to generate the most accurate set of PPI.

2.2 *Sacchraomyces cerevisiae* Protein-Protein Interaction Network

Throughout this dissertation, we will examine and analyze the *Saccharomyces Cerevisiae* PPI network as described in the Database of Interacting Proteins (DIP) (Xenarios *et al.* , 2002). More specifically, this network was described by Gavin *et al.* in 2002 through use of tandem-affinity purification (TAP) and mass spectrometry (MS). In this process, "individual proteins are tagged and used as hooks to biochemically purify whole protein complexes" (Von Mering *et al.* , 2002). One main advantage of the TAP/MS procedure is that it can detect real complexes *in vivo* as opposed to potentially artificial complexes *in vitro*. Unfortunately, it may not detect PPI that are not present during the specific physiological settings under which the test is performed. Slightly different settings may lead to the discovery of different protein complexes. In addition, the tagging may disturb complex formation causing unnatural changes or may not bond closely enough. In the latter situation, the tag may be washed off and thus the interaction will not be recorded (Von Mering *et al.* , 2002).

Saccharomyces Cerevisiae is a species of yeast used in winemaking, baking, and brewing. This was the first eukaryotic organism whose entire set of proteins and corresponding interactions was analyzed (Mashaghi *et al.* , 2004). The dataset used here has 1361 proteins (nodes) with 3222 interactions (edges). The corresponding PPI network has the majority ($\sim 92\%$) of nodes in one giant (connected) component. Table 2.1 shows the comparison, when applicable, between measures of the full network and measures of the

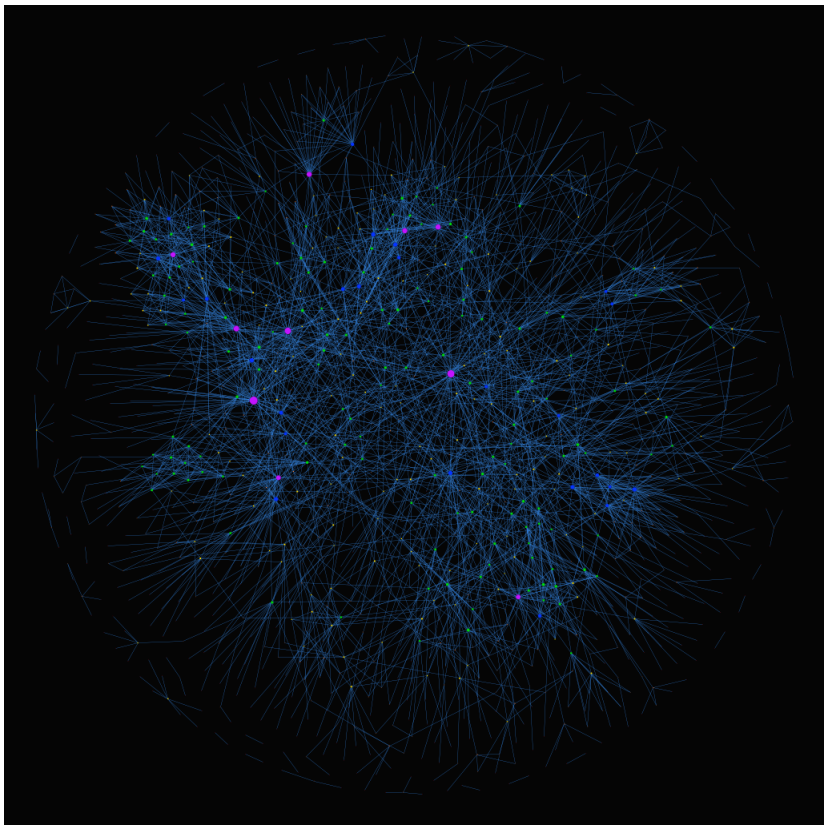


Figure 2.1. **Visualization of the *S. cerevisiae* protein-protein interaction network.**

giant component. In general, there is very little change in value between the full network and the giant component. This is a useful property of network comparisons. The dataset has no proteins interacting with themselves and no lone nodes. The network has a density of 0.0035, which is much closer to the lower end of the density range than to the upper. This is quite common in real world networks (Melancon, 2006). The transitivity (Equation 1.7) of the full network is 0.1934, therefore there are many more unconnected triads than full triangles found in the network. Finally, the assortativity coefficients of the yeast data, both for the giant component and full network, are negative. This implies dissassortative mixing: high degree nodes are preferentially attached to low degree nodes (Estrada, 2011). This is very commonly seen in biological networks and thus was to be expected (Barabási & Oltvai, 2004).

Several other measures provide good insight into the shape and size of the network. The PPI network has a diameter of 12 and a radius of 6. Its characteristic path length, or average shortest path length, is 4.8972. By using this measure along with the average

Table 2.1. Table of graph measures for the *S. cerevisiae* PPI network.

	$\mathcal{G}$	$\mathcal{H}$
n	1361	1246
m	3222	3142
$\bar{k}$	4.7348	5.0443
$\lvert V_{\mathcal{H}}\rvert/\lvert V_{\mathcal{G}}\rvert$	0.9155	-
$\mathcal{D}(\mathcal{G})$	0.0035	0.0041
$C(\mathcal{G})$	0.1934	0.1939
$\bar{C}$	0.217	0.2351
$r(\mathcal{G})$	-0.1176	-0.1441
$S(\mathcal{G})$	0.5364	0.5375
$diam(\mathcal{G})$	-	12
$rad(\mathcal{G})$	-	6
$\bar{\ell}$	-	4.8972
$\bar{DC}$	0.0035	0.0041
$\bar{CC}$	0.1749	0.2085
$\bar{BC}$	0.0024	0.0031
$\bar{\psi}$	0.0087	0.0095

$\mathcal{G}$ indicates the full network. $\mathcal{H}$ is the giant connected component of the network. Not all values be calculated for both the full network and the giant component because measures examining paths require a connected network.

clustering coefficien we can attempt to ascertain whether this network displays signs of either small-world or scale-free properties. The estimate of average shortest path length for a random graph of the same size and with the same degree is 4.64. The estimate for average clustering coefficient is 0.0034. Using the proportions seen in Equations 1.15 and 1.16, we can calculate p and q:

$$\begin{aligned} p &= \frac{\bar{C}}{\bar{C}_r} \\ &= \frac{0.217}{0.0034} \\ &= 63.82 \end{aligned} \tag{2.1}$$

$$\begin{aligned} q &= \frac{\bar{\ell}}{\bar{\ell}_r} \\ &= \frac{4.8972}{4.6437} \\ &= 1.05. \end{aligned} \tag{2.2}$$

Clearly, $p \gg 1$ and $q \approx 1$. Therefore, the *S. cerevisiae* PPI network has small-world properties.

We can also check for the presence of scale-free properties. In this situation, the average shortest path length of the random graph with the same number of nodes calculated by Equation 1.19 is 6.32. Then we can calculate s:

$$\begin{aligned} s &= \frac{\bar{\ell}}{\bar{\ell}_{r,ultra}} \\ &= \frac{4.8972}{6.3173} \\ &= 0.7752. \end{aligned} \tag{2.3}$$

From Equation 2.3 we see that $s = 0.7752$. This is clearly not approximately equal to one, thus we can conclude the *S. cerevisiae* PPI network does not exhibit scale-free features. This failure to recreate scale-free features is evidence to support the argument that PPI networks, in general, are not scale-free. The S-metric, $S(\mathcal{G})$, is another metric that aids the argument. Since the S-metric identifies a hub-like core if there is one present, we would expect a value closer to one than to zero. However, the S-metric for the full PPI network is just over 0.5 (Table 2.1). The S-metric for the giant component is essentially the same. Thus the *S. cerevisiae* PPI network does not show evidence of a hub-like core, the hallmark of the scale-free network (Li *et al.* , 2005). Despite these measures, it is still widely held that PPI networks are indeed scale-free (Jeong *et al.* , 2000; Barabási & Oltvai, 2004; Albert, 2005). This question is a main motivator of the following research. The discrepancies in classifications of PPI networks, as well as diversity in classifying methods, have not yet been adequately addressed.

2.3 Motivation for Network Classification

There are many reasons why the analysis of PPI, as well as the determination of the best fitting model graph, is important. One factor is orthologous proteins. Orthologs refer to "genes that have diverged after a speciation event" (Fulton *et al.* , 2006). Thus the encoded proteins of these ortholog genes have similar functions in different species. The identification of protein interactions in less complex organisms can then lead to the discovery of their

orthologs in more complex organisms (Jeong *et al.* , 2001). In fact, it is believed "that a significant number of the yeast complexes described [here] will have human equivalents and these may form the basis for understanding multifactorial disease" (Gavin *et al.* , 2002).

Another reason that the determination of the most accurate growth mechanism of PPI networks is important is for use in predicting missed interactions as well as identifying false-positive interactions. As was previously noted, PPI datasets are thought to have extremely high numbers of false-positive interactions. Finally, PPI network analysis can further investigations about evolutionary processes (Emmert-Streib, 2012).

In the next chapter (Chapter 3), we present the nine different types of model graphs that will be considered as possibilities to best mimic the *S. cerevisiae* PPI network. From there we move onto a numerical comparison of the model graphs to the *S. cerevisiae* PPI network based on measure values (Chapter 4). Finally, present five popular classifiers (Chapter 5) and test their performance capabilities in Chapters 6 and 7.

Chapter 3

Introduction to Model Graphs

The term *model graph* refers to any graph that is not specifically designed to model a real-world network. Nodes in the graph cannot be directly mapped to real-world entities. Instead, these graphs are built by using a prescribed algorithm, also known as a *growth mechanism*. Model graphs can be split into two mutually exclusive groups: static and growing. When static graphs are created, all of their nodes are present. Thus, there is no concept of node age. These graphs may begin with a partially connected seed graph along with numerous lone nodes or with just a collection of lone nodes. Edges are then added based on the prescribed algorithm.

The second group of graphs can be classified as growing. These models begin with a small connected seed graph, similar to the start of some of the static graphs, however there are no long nodes present. Each node is added to the seed graph during its own time step, with edges being added or removed at the same time. None of the static graphs contain a mechanism for edge removal while many of the growing ones do. Growing a model typically leads to different properties than those found in a static graph (Callaway *et al.* , 2001). One such property is node age, which is often important in modeling real-world networks.

In this chapter we introduce the nine model graph types that will be used throughout this dissertation.We begin with an overview of each algorithm, followed by simulations to create 1000 model graphs of each of the nine types. The results of the simulation are compared based on many of the graph measures presented in Chapter 1 in order to determine the amount of variability within each model type. We conclude with a full examination of this variability.

3.1 Model Graph Descriptions

A total of nine model graph types are examined in the search for the best fit for the *S. cerevisiae* PPI network. The graphs were chosen based on their usage in the methods that will be examined in the following chapters, as well as their prevalence in the literature. Su et al (Su *et al.* , 2011) analyzed the duplication-mutation-complementation (Vázquez

et al. , 2003), duplication-mutation with random mutation (Sole *et al.* , 2002), and linear preferential attachment models (Yule, 1926; Simon, 1955; Barabási & Albert, 1999). Przulj *et al.* (Przulj *et al.* , 2004; Przulj, 2007) used the Erdös-Rényi random graph (Erdős & Rényi, 1960), Erdös-Rényi random graph with specified degree distribution (Molloy & Reed, 1995), and linear preferential attachment models. The Erdös-Rényi random graph is also called a random static network. In the earlier paper, Przulj also used 2-dimensional, 3-dimensional, and 4-dimensional geometric graphs (Przulj *et al.* , 2004), while in later papers she only used the 3-dimensional geometric graph because it was found to be the best fit out of the three types of geometric networks (Pržulj & Higham, 2006; Przulj, 2007). Higham also used geometric graphs, but found that 2-dimensional geometric graph are "generally as effective as higher dimensional Euclidean space for explaining the connectivity" (Higham *et al.* , 2008). A third paper by Przulj introduced the concept of a stickiness index to design a graph based on "the abundance and popularity of binding domains on a protein" (Pržulj & Higham, 2006). Another paper, by Middendorf *et al.* (Middendorf *et al.* , 2005) utilized the most model graphs for comparison, a total of seven. These include aging vertex (Amaral *et al.* , 2000), duplication-mutation-complementation, duplication-mutation with random mutation, linear preferential attachment, random static, random growing, and small-world (Watts & Strogatz, 1998). Obviously multiple graphs, specifically the duplication models, linear preferential attachment, and random static, were used in multiple papers. This would seem to signify that scientists expect these models to have the best fit out of all the graph choices for the PPI networks.

As previously mentioned, the nine graph types can be split into two mutually exclusive groups. The first group is composed of static graphs, which lack the concept of node age. Models falling into this category include random static, small-world, and geometric (Table 3.1). The second group of model graphs can be classified as growing (Krapivsky *et al.* , 2000; Dorogovtsev *et al.* , 2000). Models in this category include random growing, duplication-mutation with complementation, duplication-mutation with random mutation, aging vertex, linear preferential attachment, and stickiness index (Table 3.1). These graphs have the concept of node age, which is often important in modeling real world networks.

Table 3.1. Model graphs used for network classification.

Model Graphh	Abbrev.	Type
3D Geometric	GEO	Static
Random Static	RDS	Static
Small-World	SMW	Static
Aging Vertex	AGV	Growing
Duplication-Mutation-Complementation	DMC	Growing
Duplication-Mutation with Random Mutation	DMR	Growing
Linear Preferential Attachment	LPA	Growing
Random Growing	RDG	Growing
Stickiness Index	STI	Growing

This table shows the nine model graph types used as potential best fits for *S. cerevisiae* PPI network classification throughout this dissertation. The first column provides the name of the model graph. The second column is the abbreviation that will be used throughout this work. Finally, the third column indicates whether the graph is growing or static. Static graphs begin with all of their nodes present and edges are added based on the algorithm. Growing graphs begin with a seed graph and new nodes and edges are added at different time-steps.

3.1.1 Random Static

The first and most basic model type considered is the random static graph (RDS), also referred to as the Erdös-Rényi random graph. It begins with a completely unconnected graph of n nodes (Middendorf *et al.* , 2005). Two vertices are randomly chosen and connected. This is repeated until the number of edges reaches the desired number. The RDS network results in a Poisson degree distribution (Callaway *et al.* , 2001).

3.1.2 Small-World

Another very common graph type that is utilized is the small-world graph. This model was first discussed by Watts and Strogatz (Watts & Strogatz, 1998) and is characterized by its short characteristic path length, $\bar{\ell}$, and high degree of clustering. Neither of these properties can be captured with traditional approximations based on lattices or random graphs, which led to the motivation for creating this type of model graph. In short, small-world graphs have "short cuts" which increase the connectivity, leading to the "six degrees of separation" phenomena (Barrat & Weigt, 2000). The characteristic path length can be approximated by $\bar{\ell} \sim \log n$ (Cohen & Havlin, 2003).

The small-world graphs for this analysis begin with a regular ring lattice of 1361 nodes with every node connected to its neighbors at a maximum distance $\left[\frac{m}{n}\right]_- = 2$ with a

probability of:

$$\left[\frac{m}{n}\right]_+ - \frac{m}{n} = 3 - 2.3674$$
$$= 0.6326.$$

In addition, nodes are connected to neighbors at distance of $\left[\frac{m}{n}\right]_+ = 3$ in order to make the average total number of edges equal to m, which in this case is 3222. Rewiring of the graph occurs by randomly selecting a pair of edges (v_i, v_j) and choosing another vertex, v_k that is not connected to v_i. The rewiring of (v_i, v_j) to (v_i, v_k) is performed with probability $q_{rewire} \in (0, 1)$ (Middendorf *et al.* , 2005).

3.1.3 3D-Geometric

A geometric random graph, $G(n, r)$ is a geometric graph containing n nodes and radius, r (Pach, 1999). The n nodes correspond to n "independently and uniformly randomly distributed points in a metric space" (Przulj *et al.* , 2004). Here we consider 3-dimensional geometric graphs with a corresponding metric space of $[0, 1]^3$ (Penrose, 2003). The 3-dimensional geometric graph was chosen over other geometric graphs based upon the results achieved by Przulj *et al.*, who show that it is a better fit to the PPI network examined (Przulj *et al.* , 2004).

3.1.4 Linear Preferential Attachment

The linear preferential attachment (LPA) model by Barabasi and Albert (Barabási & Albert, 1999) is based on the idea that when a new vertex attaches to a graph it prefers to attach to vertices that are already well connected with a probability proportional to $k + a$, where a is a constant and k is the degree of the node. Thus a higher level of connectivity leads a node to obtain more of the new additions than a node with a lower level. This property, along with the constant expansion of a network due to new additions leads to the scale-free property. In addition, scale-free graphs are ultra-small. This means that their characteristic path length can be approximated by $\bar{\ell} \sim \log\log n$ (Cohen & Havlin, 2003).

The LPA model, along with the next several model types, each begin with a seed RDS graph composed of $[2\frac{m}{n}+1]_+$ nodes and $\frac{m}{n}[2\frac{m}{n}+1]_+$ edges (Middendorf *et al.* , 2005). Since all model graphs are based on the *S. cerevisiae* PPI network with 1361 nodes and 3222 edges, the number of nodes in the seed graph is 6 and the number of edges is approximately 14. The constant $a \in (0, 5)$, with the upper limit chosen based on prior research and trials performed by Middendorf *et al.* From there preferential attachment is used to build edges between the nodes (Middendorf *et al.* , 2005).

3.1.5 Random Growing

The random growing graph, RDG, begins with the same RDS seed graph setup of 6 nodes and 14 edges as the LPA graph (Middendorf *et al.* , 2005). At each time step, a new vertex is added to the list of nodes in the graph, but it is not necessarily connected to the graph immediately. Two nodes are randomly chosen and connected. The process of connecting nodes is repeated until the total number of edges added at that time step is greater or equal to:

$$\left[\frac{m}{n}\right] = 2.3674. \tag{3.1}$$

The addition of edges continues until the desired number has been achieved. Despite beginning with an RDS graph, the RDG graph is distinctly different (Krapivsky & Redner, 2001). The difference likely results from the fact that in a growing graph vertices have distinct ages, a property not found in the RDS network. The group of older edges has more time to develop interactions and thus form a tight core that has a higher than average density of edges. This core leads to the illusion that highly connected nodes are more likely to be connected to each other, or the illusion of preferential attachment. In addition, despite the fact that preferential attachment is absent, the growth of a graph, as opposed to its creation, causes clearly identifiable differences in characteristics (Callaway *et al.* , 2001). Thus the growing random graph has different characteristics than the random static graph. For example, the resulting degree distribution of the RDG graph is exponential, as opposed to Poisson for RDS.

3.1.6 Aging Vertex

The aging vertex graph, or AGV, is based upon graph models of citation networks, the Internet, and scientist collaboration networks (Klemm & Eguiluz, 2002; Zhu *et al.* , 2003). All of these networks display preferential attachment, where the likelihood of a node obtaining a new link is directly proportional to the number of links that it already has. These networks also have an additional property that is seen most clearly in the scientist collaboration network (Newman, 2001a; Barabâsi *et al.* , 2002; Moody, 2004); after a period of time, a scientist will have no more new collaborations because they will no longer be active. Thus, after a certain period of time a node receives no more edges, however many it already has. A similar phenomenon is seen in the citation network (Chen & Redner, 2010), where papers become less citepd as they become older and outdated. The AGV graph is designed to represent these occurrences by containing three distinct empirical properties. First, the degrees follow a power law distribution. Second, preferential attachment is utilized for the addition of new nodes. Third, there is a negative correlation between age and the addition of new links (Klemm & Eguiluz, 2002). Beginning with the seed graph, nodes are chosen randomly and connected based upon properties of other nodes.

3.1.7 Duplication-Mutation-Complementation and Duplication-Mutation using Random Mutation

The duplication-mutation-complementation graph model (DMC), also known as the duplication-mutation preserving complementarity graph, and the duplication-mutation using random mutation graph (DMR) are both biologically based. The DMC model was first described by Vazquez and Flammini in 2003 (Vázquez *et al.* , 2003). The graph begins with a path of length two, two nodes connected by one edge. At each time step a new node is added to the graph, v_{new}. It chooses another node at random, v_{old}, and copies all of its neighbors. For each neighbor of the two nodes (they currently have identical neighbors), the edge connecting it to either v_{new} or v_{old} is randomly selected. This selected edge is then removed with a probability $q_{del} \in [0, 1]$. Finally, v_{new} is connected to v_{old} with probability $q_{con} \in [0, 1]$. The biological implication of only removing one edge to any given neighbor is that it allows for the preservation of function. Finally, v_{new} is connected to v_{old} with probability $q_{con} \in [0, 1]$

(Su *et al.* , 2011; Middendorf *et al.* , 2005). This process continues until the designated number of nodes has been added.

The DMR graph was first described in 2002 by Solé (Sole *et al.* , 2002). It begins similarly to the DMC graph, but the seed graph is a 5-vertex-cycle, $|C_5|$ (Middendorf *et al.* , 2005). At each time step a new node, v_{new}, copies all of the neighbors of a previously connected node, v_{old}. For each of those neighbors, the probability that its edge to v_{old} is deleted with probability $q_{del} \in [0, 1]$. A link between the new node and any of these nodes is created with probability $q_{new}/(n_t - 1)$, where $q_{new} \in [0, 1]$ and $(n_t - 1)$ is the total number of nodes in the graph, not including v_{new}, at time step t. Thus this model allows for completely new interactions that the DMC model does not (Su *et al.* , 2011). Once again, the process continues until the designated number of nodes has been added. Both biologically based graphs lead to far larger variations in the numbers of nodes and edges than any other model examined here because of their propensity for lone nodes, which are removed and not counted in this analysis, as well as a lack of constraint on the number of desired edges.

3.1.8 Stickiness Model

The final model, the stickiness model (STI), is also biologically based. It was proposed by Przulj *etal*. in 2006 (Pržulj & Higham, 2006) and employs a stickiness index. This index is based upon the normalized degree of a node. The purpose of this model is to mimic the binding domains found on proteins. The model is motivated by two assumptions. First, it assumes that a protein (node) with a high degree has many binding domains and/or its domains are commonly involved in interactions. The second assumption is that two proteins are more likely to interact (have an edge) if they both have high stickiness indices. In previous tests this model has been found to be the best fit for 14 PPI networks derived from different species at different levels of confidence with 25 comparison models (Pržulj & Higham, 2006). The STI graph has also been shown to be the best fit for viral PPI networks (Kuchaiev *et al.* , 2011).

3.2 Methods

For each of the nine model types, 1000 graphs were created. We chose to simulate 1000 graphs because it is the most common number of simulations run across numerous simulation studies (Burton *et al.* , 2006). The graphs for AGV, DMC, DMR, LPA, RDS, RDS, and SMW were created using source code provided by Middendorf *et al.* Pseudo-code for these calculations can be seen in the Supplemental Information of their paper (Middendorf *et al.* , 2005). The programs building these model graphs were run in MATLAB (MATLAB, 2010). The remaining graphs were created using GraphCrunch 2 (Kuchaiev *et al.* , 2011). In the creation of each type of model graph, either the number of nodes, the number of edges, or both were required to be specified. For these values, the numbers of nodes and edges of the *S. cerevisiae* PPI network were used as input. It should be noted that even though we use those value as inputs, it does not guarantee that each model graph created will have the exact criteria desired. Many of the growth mechanisms resulted in lone nodes that were removed, a feature built into the code used and consistent across the literature (Przulj *et al.* , 2004; Middendorf *et al.* , 2005; Pržulj & Higham, 2006; Przulj, 2007; Su *et al.* , 2011). In addition, multiple mechanisms have a randomly sampled probability of edge creation, as opposed to a specified number of edges, and thus the resulting graphs can differ drastically from the ideal.

The 1000 model graphs of each type were assessed based on fifteen measures: average shortest path length, assortativity, average clustering coefficient, average degree, betweenness centrality, closeness centrality, degree centrality, density, diameter, number of edges, eigenvector centrality, proportion of nodes in the giant component, number of nodes, radius, and transitivity. Results for each measure are illustrated with box plots, allowing us to examine the distribution within each model type as well as to compare against other types.

3.3 Results

3.3.1 Numbers of Nodes and Edges

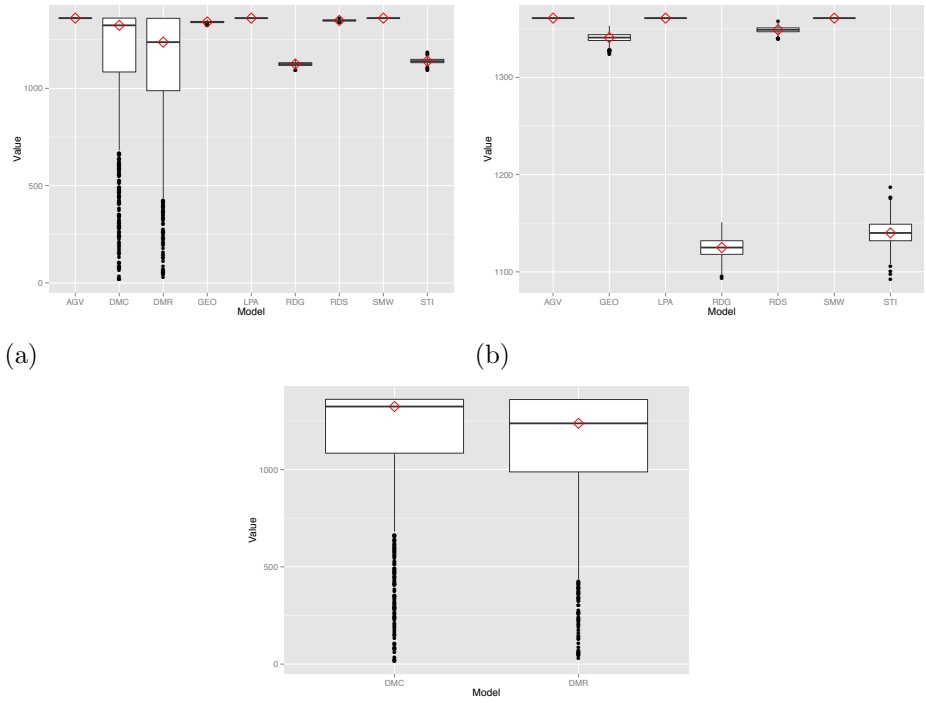
We begin by examining the numbers of nodes (Figure 3.1) and edges (Figure 3.2) within each model. It was previously noted that for the majority of models, both the number

of nodes and the number of edges were entered into the algorithm building the graph. A literature review indicated that lone nodes were most commonly dealt with by eliminating them from the graph, thus leaving the model graph with less nodes that desired. We decided to continue this trend for two reasons. First, it is necessary to keep as many aspects of the simulation the same as in the literature so that results can be properly compared. Keeping lone nodes, or continuing to run the model until the ideal number of nodes are connected changes the experiment and thus will logically change the results of the classification. This would prevent us from being able to make a true comparison. Second, all of the classifiers later examined have the ability to deal with differences in number of nodes, especially when the largest difference is still less than a 20% reduction. Finally, protein-protein interaction networks are, obviously, built on interactions. Model types that tend to build graphs with a huge number of lone nodes are most likely not the best fitting model anyway.

Figure 3.1a, shows a plot of all of the model types. Node that the DMC and DMR graph types have a huge range in the number of nodes. This indicates two things. First, they are clearly much more prone to the creation of lone nodes than the other model types judging from the lower end of the range of values. Second, they do not produce as consistent of a model graph as the other growth mechanisms based on the absolute size of the range.

Figure 3.1b shows a closeup of Figure 3.1a but with the DMC and DMR graphs removed. At this closer level, we can see that RDG and STI graphs also tend to produce more lone nodes and more varied graphs than the remaining five models. Figure 3.1c, shows a plot of the number of nodes in DMC and DMR graphs alone. DMC graphs have a higher median number of nodes and a larger range than the DMR graphs.

For number of edges in the model graphs, two model types did not take this number as input: DMC and DMR. Similar to the issues seen in the number of nodes, we chose not to modify the algorithms for either of these models at this point largely for the sake of retaining comparisons to previous works. The DMC and DMR models were both designed to mimic PPI networks (Sole *et al.* , 2002; Vázquez *et al.* , 2003) and several papers have found them to be the best out of all the models tested (Middendorf *et al.* , 2005; Su *et al.* , 2011). Therefore, we decided to keep these growth mechanisms as is.



(a) (b)

(c)

Figure 3.1. **Comparison of the number of nodes across model graph types.** Each box plot shows the distribution of the number of nodes for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding DMC and DMR. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of DMC and DMR model graphs.

Figure 3.2 shows three plots demonstrating the number of edges in each of the model graphs. It is obvious, once again that DMC and DMR graphs produce a huge range of edges in each graph (Figure 3.2a). In fact, when all of the models are shown on the same subplot, it is impossible to obtain any information about the other model types because the DMC and DMR graphs dominate the image. When these two graphs are removed, Figure 3.2b shows that STI also demonstrates a significantly large range in number of edges compared to the other models. When just DMC and DMR are shown, the lower end of the range for both models is approximately the same, as is the median, however the DMC growth mechanism produces graphs with a significantly larger number of nodes that DMR.

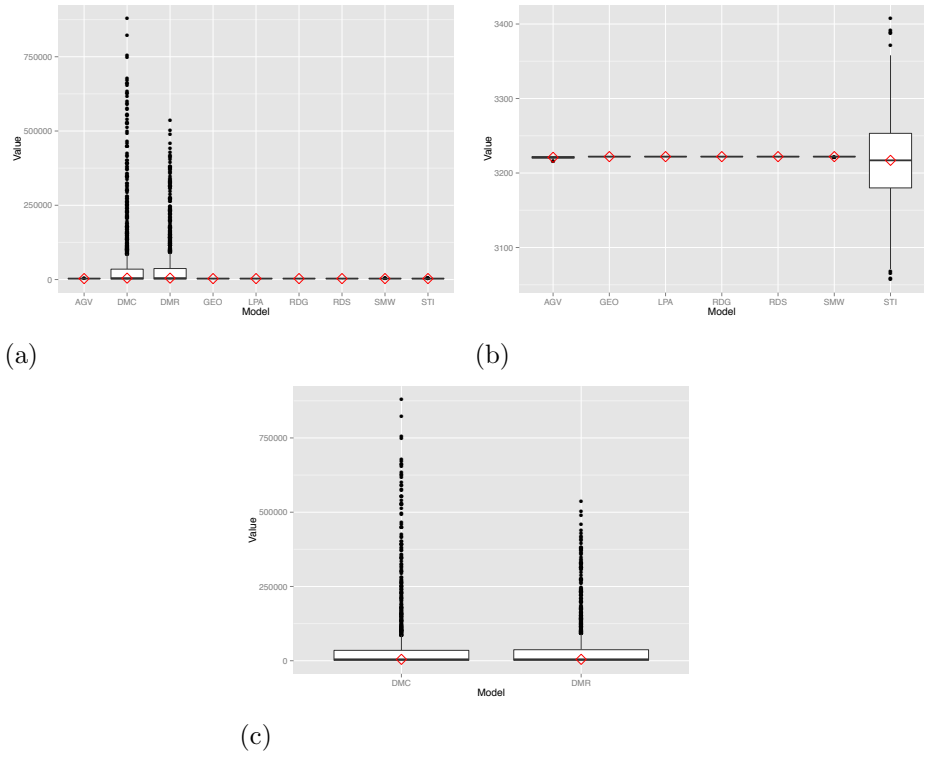


Figure 3.2. **Comparison of the number of edges across model graph types.** Each box plot shows the distribution of the number of edges for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding DMC and DMR. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of DMC and DMR model graphs.

3.3.2 Density

The density of a graph is the proportion of edges divided by the proportion of possible edges (Equation 1.3). The same pattern that we saw for the number of nodes and number of edges in the model graphs shows itself here as well; graphs produced by the DMC and DMR growth mechanisms are much more varied than the others. Once again, when the box plots for all of the model graphs are shown in the same frame, these two model types dominate the image making it difficult to obtain any information about the other model types (Figure 3.3a). When shown alone (Figure 3.3c) DMC has a larger range of values than DMR though the bottom of both ranges and median values are approximately the same.

The density of the remaining model graphs plotted without DMC and DMR shows that RDG and STI graphs have the next two most varied spreads of density. This follows from the results seen in Figures 3.1b, 3.2b. RDG and STI graphs have greater variations in their numbers of nodes than the other model types, with the exception of DMC and DMR. STI also has greater variation in its number of edges. Since density is a function of nodes and edges, larger variations in those two measures will logically lead to the larger variation seen in Figure 3.3b.

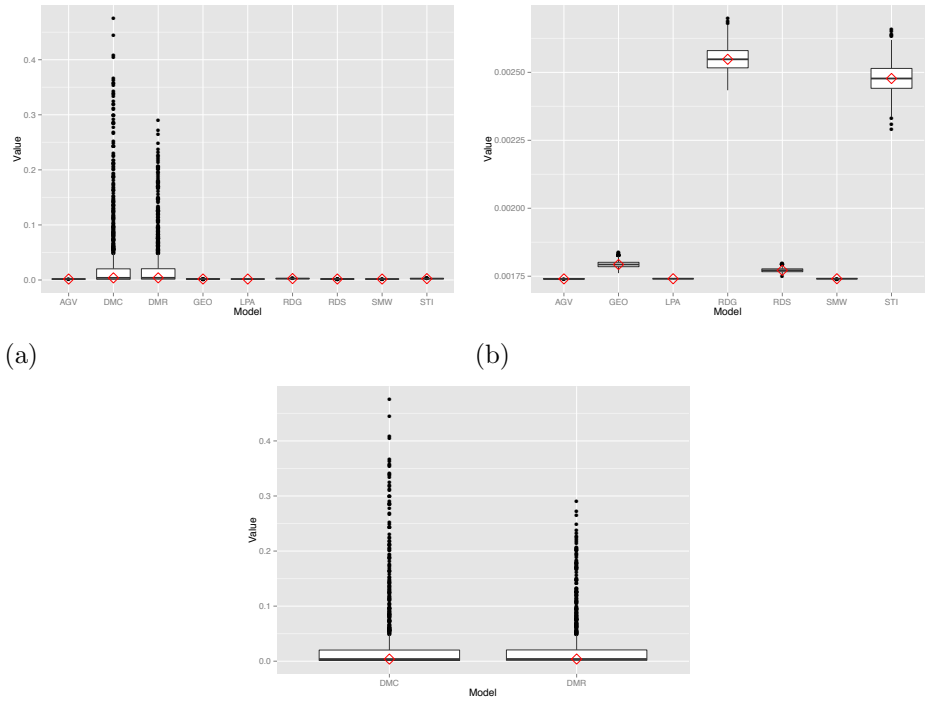
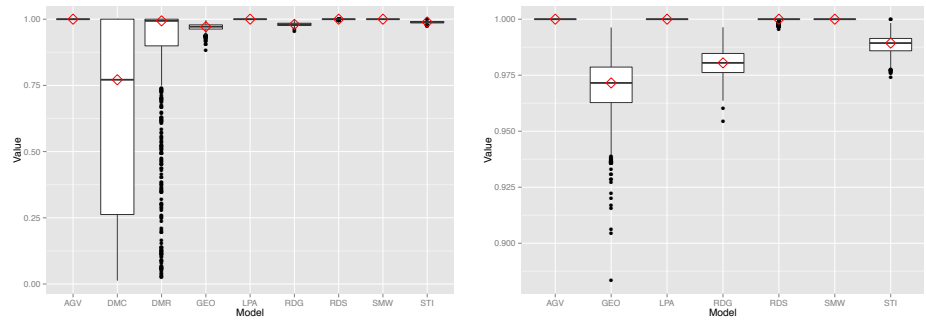


Figure 3.3. **Comparison of the graph density across model types.** Each box plot shows the distribution of the density for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding DMC and DMR. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of DMC and DMR model graphs.

3.3.3 Proportion of Nodes in the Giant Component

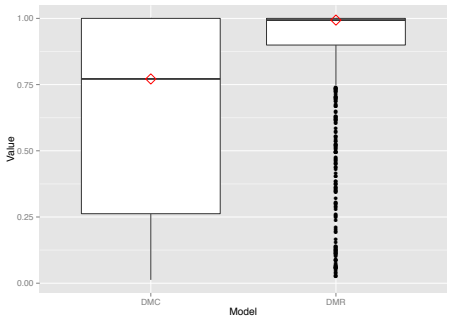
The giant component of a graph is the largest connected component of the graph. The proportion of nodes in the giant component is the number of number in the giant component divided by the total number of nodes. The majority of graphs have proportion greater than 90% (Figure 3.4a. This is even true for DMR graphs. Despite their long tailed distribution, the median is just under 100% (Figure 3.4c). DMC on the other hand has a median closer to 75% and its IQR range dwarfs the other model types.

When DMC and DMR are removed from the plot, allowing a more concentrated view of the remaining model types, the varied nature of the size of the connected component for GEO, RDG, and STI becomes more clear. GEO graphs do not contain much variation in either their numbers of nodes or edges. This indicates that while these graphs are not prone to producing lone nodes, they are prone to producing smaller components that do not connect to their larger one. Since they are not prone to producing lone nodes, we can infer that the size of these components is slightly significant. RDG and STI graphs are prone to producing lone nodes, so it makes sense to infer than the nodes not in the giant component are probably clustered in small groups.



(a)

(b)



(c)

Figure 3.4. **Comparison of the proportion of nodes in the giant component across model graph types.** Each box plot shows the distribution of the proportion of nodes in the giant component for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding DMC and DMR. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of DMC and DMR model graphs.

3.3.4 Diameter, Radius, and Average Shortest Path Length

Diameter, radius, and average shortest path length (ASPL) all deal with path lengths and thus all require a connected network. Therefore, these three measures were all calculated on the giant component of each model graph. Diameter, or the maximum eccentricity of a network, is always an integer. This is one of the few measures where the values expressed by DMR graphs are not greatly varied compared to those expressed by the other model types (Figure 3.5a). In fact, AGV, DMC, and SMW are graphs with substantially varied diameters. In Figure 3.5c, we see that the IQR regions are not that substantial, but that all three of these graphs have a considerable number of graphs with large, outlying values.

In Figure 3.5 we see that GEO has a significantly larger diameter than all of the other model types. This is clear even when all of the graph types are displayed together in Figure 3.5a. LPA, RDG, RDS, and STI graphs all have small IQR with just a few larger outlying diameters. There is not much variation for these model types.

Radius is the minimum eccentricity. Overall, there appears to be less variation in radius than diameter. Interestingly, the model type with the most variation is not DMC or DMR for this measure, but SMW (Figure 3.6a). When only SMW is focused on, it is clear that the IQR is not very large, however there is one graph in particular with an abnormally high radius (Figure 3.6c).

Examining the remaining model types without SMW in the plot shows that AGV and DMC have quite a few graphs with larger radii that are outliers, similar to the results for diameter (Figure 3.6b). DMR graphs do not have many outlying radii, however they do have a larger IQR than all of the other models except DMC. GEO graphs have a larger median radius than all of the other models. Finally, LPA, RDG, RDS and STI graphs have very similar median radii and do not appear to show much variation between graphs of the same type.

The box plots for ASPL look very similar to those for diameter (Figure 3.7). AGV, DMC, and SMW have the largest amount of variation. When they are pulled out and placed into their own subplot, we see more clearly that DMC has the largest IQR, but SMW has the most variance in its upper quartile (Figure 3.7c). When we look at the remaining six model types, we continue to see similar patterns. DMR has a larger IQR than the other

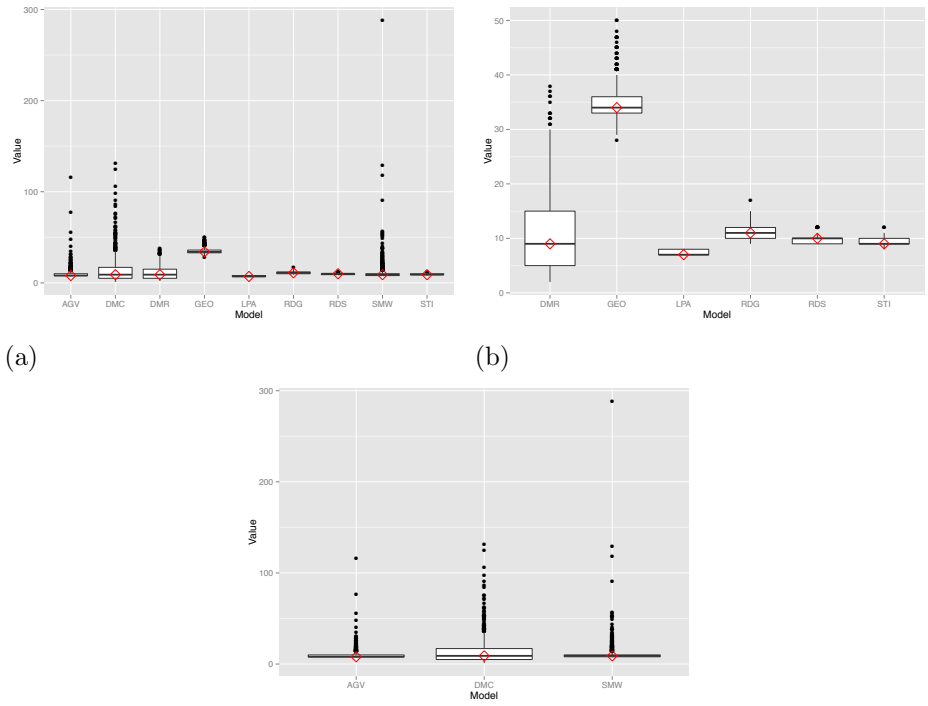
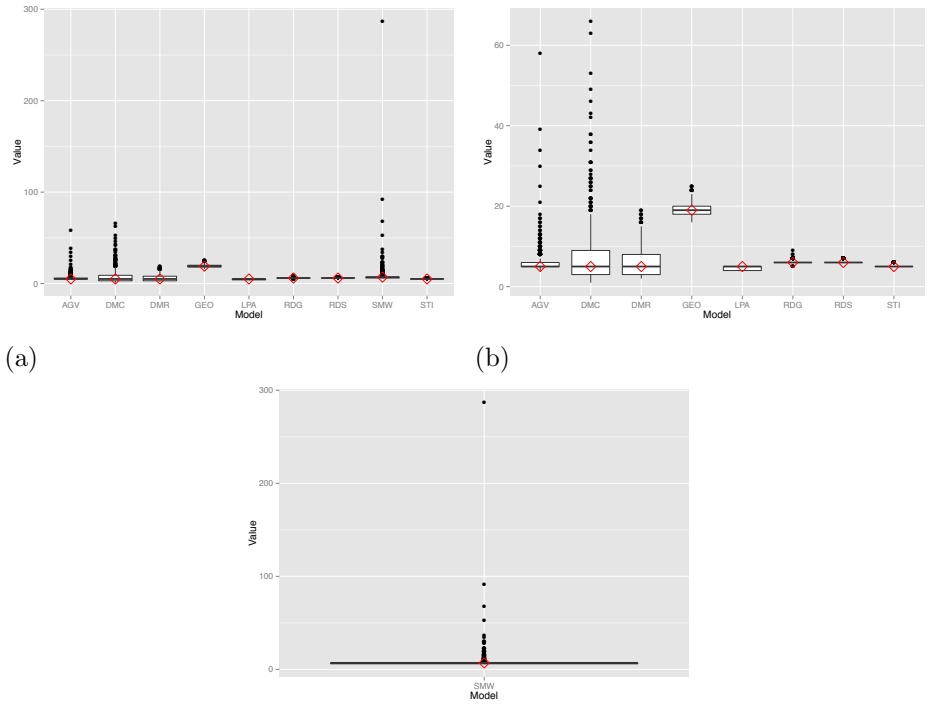


Figure 3.5. **Comparison of the graph diameter across model types.** Each box plot shows the distribution of the diameter for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding AGV, DMC, and SMW. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of AGV, DMC, and SMW model graphs.

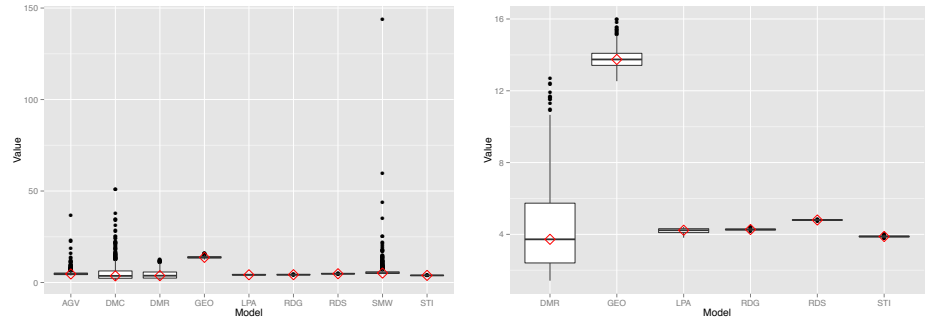
remaining models and GEO has the largest median of all of the model types including AGV, DMC, and SMW (Figures 3.7b, 3.7a). There is not much variation in ASPL for LPA, RDG, RDS, or STI.



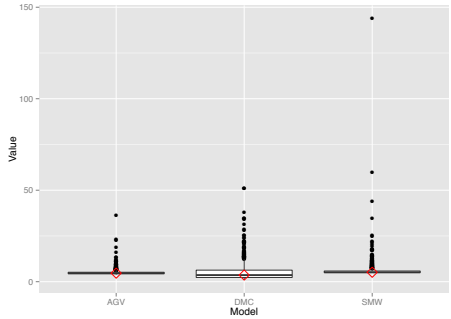
(a) (b)

(c)

Figure 3.6. **Comparison of the graph radius across model types.** Each box plot shows the distribution of the radius for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding SMW. These models were removed because their variation eclipsed the variation of the other model types. **(c):** SMW graphs distribution of radii.



(a) (b)



(c)

Figure 3.7. **Comparison of the graph average shortest path length across model types.** Each box plot shows the distribution of the average shortest path length for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding AGV, DMC, and SMW. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of AGV, DMC, and SMW model graphs.

3.3.5 Average Degree and Assortativity

Looking at the box plots of average degree for all of the model graphs it is immediately obvious that DMC and DMR graphs have much more variation than any other model type. In fact, from Figure 3.8a it is nearly impossible to determine whether any of the other models have any variation at all. Removing DMC and DMR graphs from the plots shows us that RDG and STI graph both have obvious variation in average degree across their 1000 graphs (Figure 3.8b). The remaining model types all have very limited differences between their graphs. This is particularly true of LPA and SMW. The median average degree of all of the graphs except DMC and DMR differs by less than 1.5 degrees.

Looking more closely at the average degrees of DMC and DMR, we see that the DMC values range from near zero to well over 1000 and DMR go from near zero to about 750 (Figure 3.8c). These average degrees seem impossibly large until the distribution of the number of edges is considered (Figure 3.2c). The number of edges ranged well into the upper hundred-thousands making average degrees in the upper hundreds far more plausible.

Assortativity is the likelihood that nodes with like degrees are connected. It ranges from an upper bound of one, where high-degree nodes are only connected to high-degree nodes and low-degree nodes to other low-degree nodes, down to a lower bound of negative one, where high-degree nodes are only connected to low-degree nodes. Figure 3.9 shows that the majority of the model types have an assortaitivity value near zero, which indicates that there is no preference attachment based on degree. Only two model types do not have values near zero, GEO and RDG. Both of these values are larger than zero, closer to 0.5. Of the model types hovering around zero, AGV, DMR, RDS, SMW, and STI are all just below zero and DMC and LPA are just above zero. DMC and DMR graphs still have much larger spreads of ranges than the other models. The majority of their outliers are less than zero.

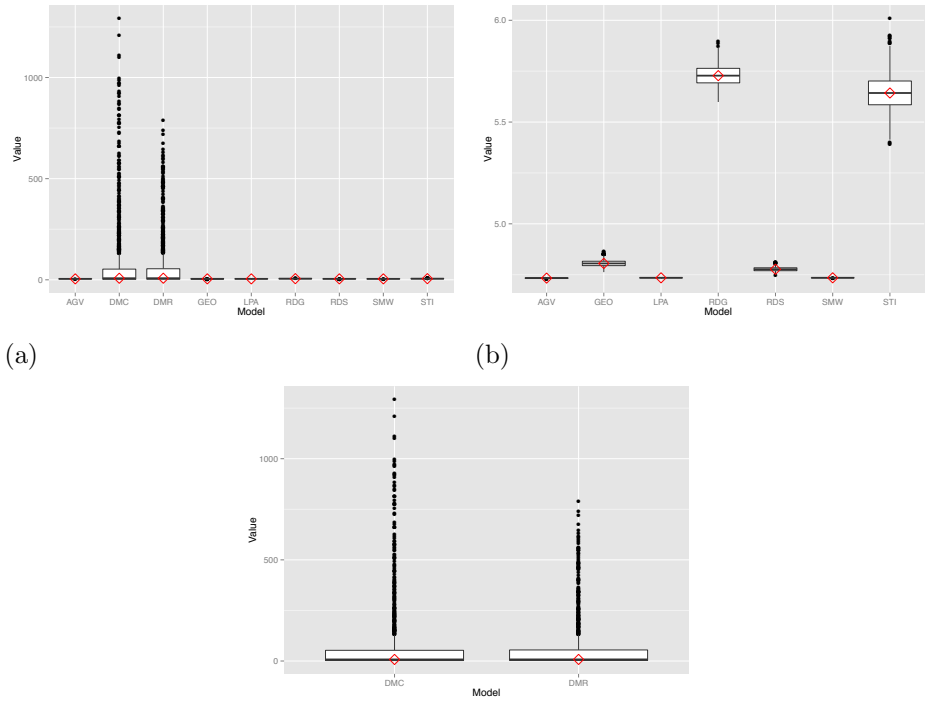


Figure 3.8. **Comparison of the graph diameter across model graph types.** Each box plot shows the distribution of the average degree for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding DMC and DMR. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of DMC and DMR model graphs.

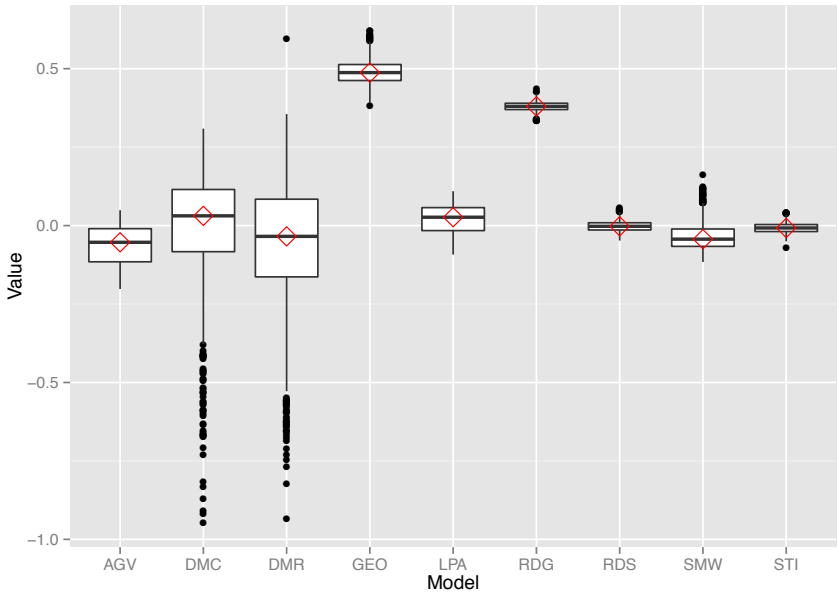


Figure 3.9. **Comparison of the graph assortativity across model types.** Each box plot shows the distribution of the assortativity for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding AGV, DMC, and SMW. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of AGV, DMC, and SMW model graphs.

3.3.6 S-metric

The S-metric is a normalized metric that determines the degree to which hub nodes in the network are connected (Beichl & Cloteaux, 2008). A value nearing one means that the hubs are connected to each other and indicates a scale-free network. A value closer to zero means that hubs are not connected and indicates a scale-rich network (Li *et al.* , 2005). In Figure 3.10, we see that four model types present with a large range of S-metric values: AGV, DMC, DMR, and LPA. AGV appears to have the largest IQR and DMC has the largest overall range. DMC also has the most outliers, all of which fall significantly below the median. GEO, RDG, and RDS graphs all present very little difference in values across models. SMW and STI graphs have some variation, but it is substantially smaller than the variation found for AGV, DMC, DMR, or LPA. GEO and SMW have the highest S-metric values while LPA and AGV have the lowest.

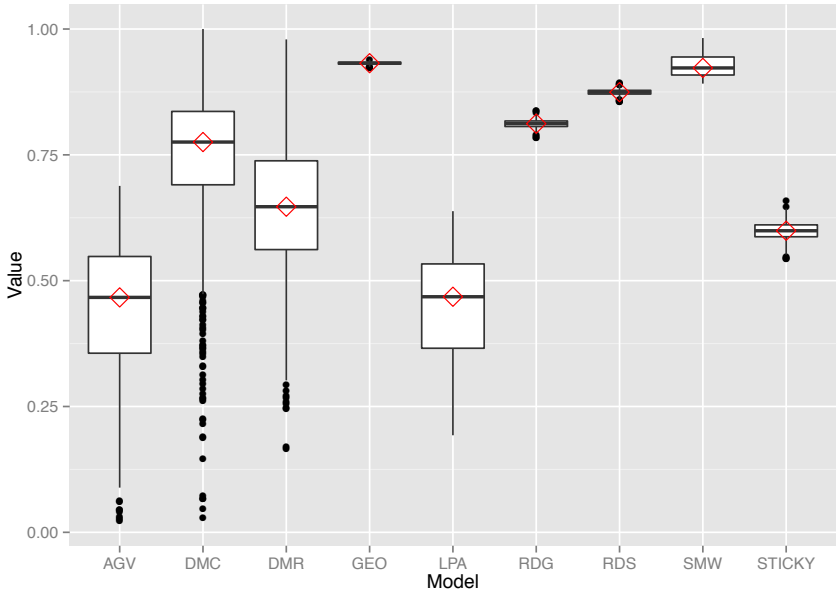


Figure 3.10. **Comparison of the transitivity across model graph types.** Each box plot shows the distribution of the S-metric for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances.

3.3.7 Clustering

There are two forms of clustering measures: average clustering coefficient and global clustering coefficient. There are two different groups of graphs based on average clustering coefficient (Figure 3.11a). There are the model types that display a large IQR along with numerous outliers and those that show very little variation. The graphs in the latter group are GEO, LPA, RDG, RDS, and STI. All of these model types, except for GEO, are shown in Figure 3.11b. GEO is not in this figure because its median clustering coefficient is significantly larger than the others in that subplot and thus adding GEO in alters the scale and provides confusion. Of the model types shown in Figure 3.11b, LPA has the largest IQR and the most outliers. STI has second largest IQR and the largest median average clustering coefficient.

The models in Figure 3.11c all have more spread out distributions of average clustering coefficient than those seen in Figure 3.11b, with the obvious exception of GEO. Of these, DMC has the second highest IQR, GEO has the highest, and most larger outliers.

Transitivity is another name for the global clustering coefficient. There is less variation across models for the global clustering coefficient than there is for the average clustering coefficient (Figure 3.12). GEO has the highest transitivity just like it had the highest average clustering coefficient. In fact Figures 3.11a and 3.12 look almost identical with the exception of RDG and LPA. LPA has a slightly higher average clustering coefficient than RDG, and RDG has a slightly higher transitivity than LPA.

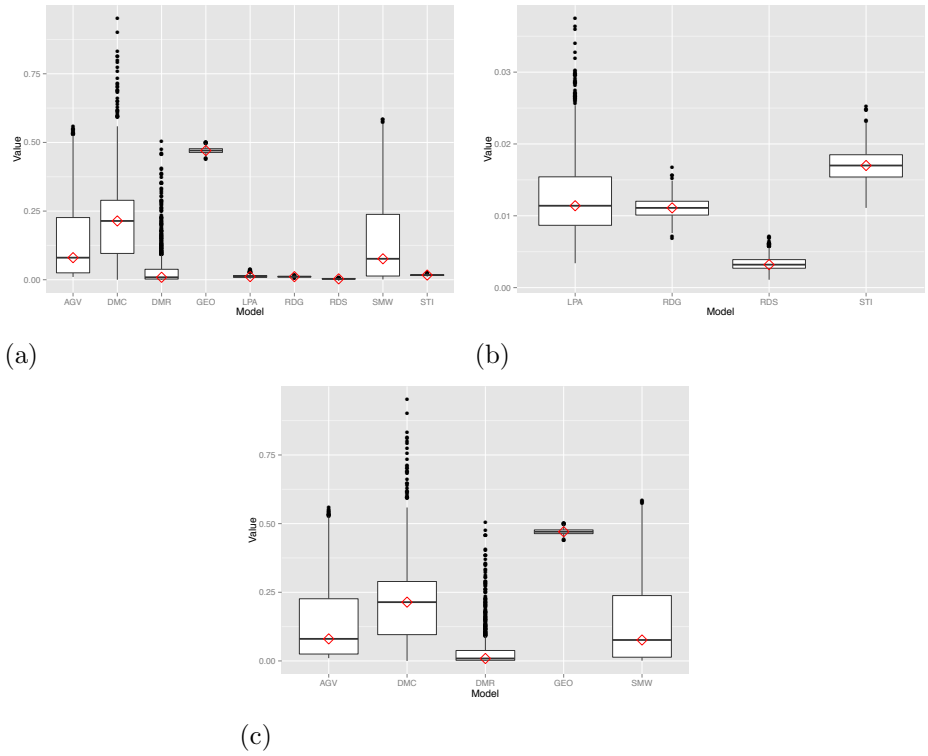


Figure 3.11. **Comparison of the average clustering coefficient across model graph types.** Each box plot shows the distribution of the average clustering coefficient for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding AGV, DMC, DMR, GEO, and SMW. These models, with the exception of GEO, were removed because their variation eclipsed the variation of the other model types. GEO graphs were excluded because their median is significantly larger than the others. **(c):** Comparison of AGV, DMC, DMR, GEO, and SMW model graphs.

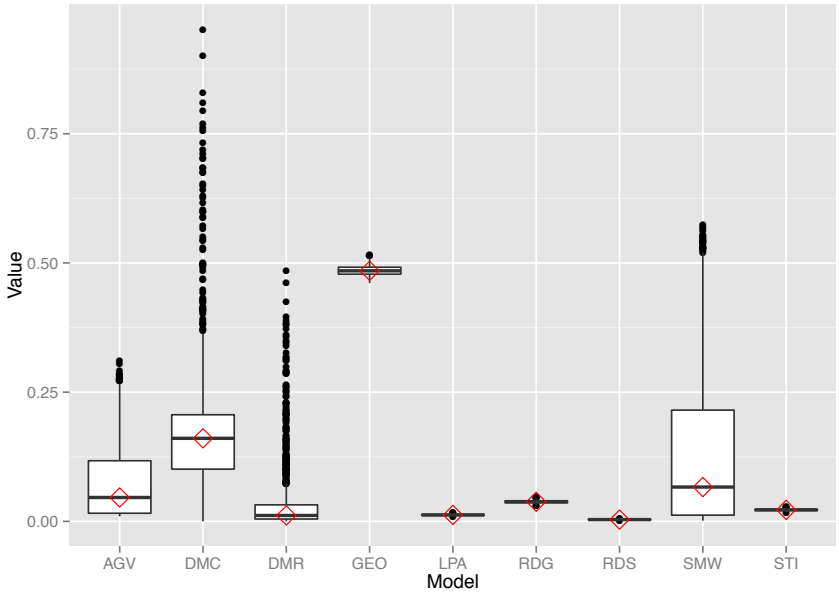


Figure 3.12. **Comparison of the transitivity across model graph types.** Each box plot shows the distribution of the transitivity for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances.

3.3.8 Centralities: Betweenness, Closeness, Degree, and Eigenvector

Centrality measures look at how important, or central, a node is in a graph. There are numerous ways to do that, but here we simply look at betweenness, closeness, degree, and eigenvector centrality. All of the values presented here are average centralities since the measure is a nodal one and we are looking to describe the full graph with only one value.

Betweenness

Betweenness centrality looks at how many shortest paths from one node to any other, go through the node in question. Figure 3.13a shows that median values do not differ much between model types. DMR has the largest IQR and SMW has the overall maximum centrality value along with the biggest range of values. Separating out SMW into its own subplot (Figure 3.13c) reduces the plot ranges and reveals a more informative picture (Figure 3.13b). In the latter picture, we see that the first three models, AGV, DMC, and DMR have much more variation in their betweenness centralities than the other model types. There is very little difference across the 1000 graphs for LPA, RDG, RDS, and STI for betweenness centrality.

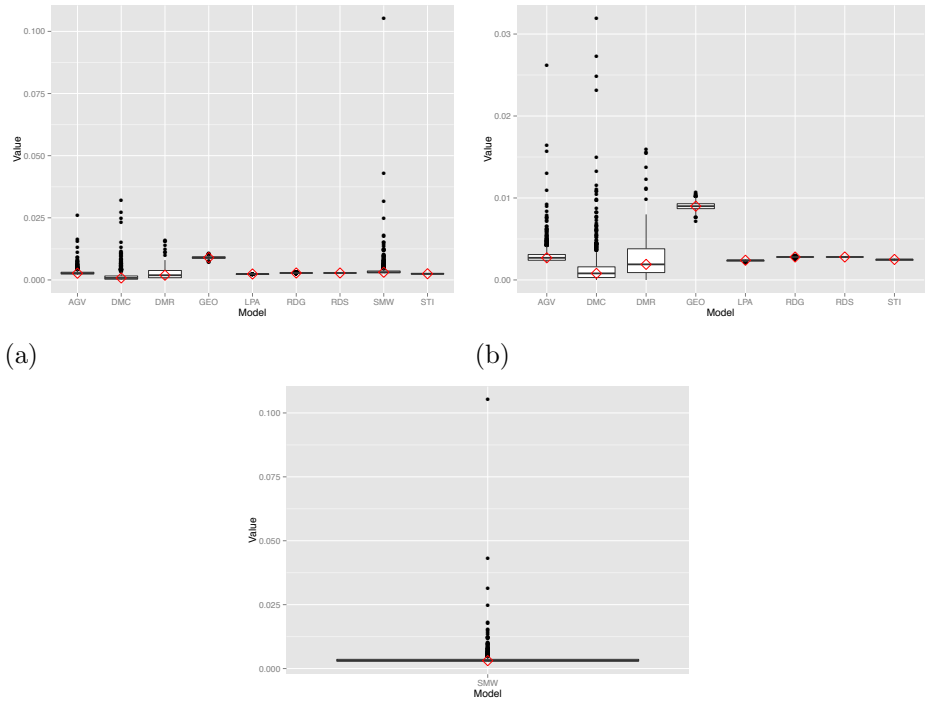


Figure 3.13. **Comparison of the average betweenness centrality across model graph types.** Each box plot shows the distribution of the average betweenness centrality for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding SMW. These models were removed because their variation eclipsed the variation of the other model types. **(c):** SMW graphs distribution of average betweenness centrality.

Closeness

Average closeness centrality is the normalized inverse of the average shortest path length for each node averaged over all of the nodes in the graph (Bavelas, 1950). In Figure 3.14 we see that DMC and DMR have significantly larger interquartile ranges than the other model types, however they are not so large that they need to be segregated into their own plot. AGV and SMW both have a significant number of outlying values smaller than their medians. GEO, LPA, RDG, and RDS do not as much variation across their graphs.

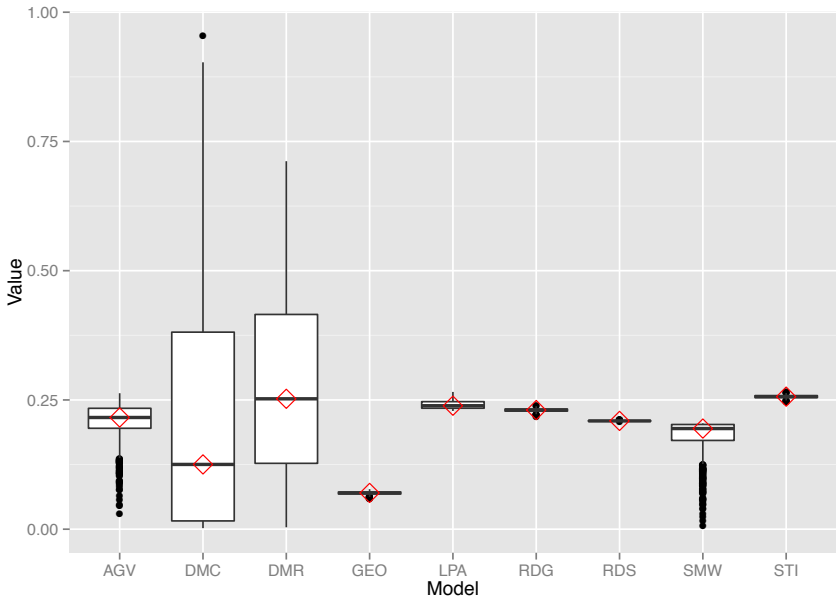


Figure 3.14. **Comparison of the average closeness centrality across model graph types.** Each box plot shows the distribution of the average closeness centrality for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances.

Degree

Degree centrality is a normalized measure of individual node degree. For this measure, we see an image virtually indistinguishable from those presented for average degree (Figure 3.8, Figure 3.15a). In Figure 3.15a, we see that the DMC and DMR model graphs display significantly more variation than all of the other model types. In fact, when the values for all of the models are displayed together, no other model type appears to have any substantial variation. Removing DMC and DMR to their own plot reveals that they have very similar medians and IQR (Figure 3.15c). Figure 3.15b shows the box plots for the remaining model types. Similar to average degree, RDG and STI have significantly higher values than the other model types. STI graphs, however, appear to have more variation as well as appearing skewed towards larger values. RDS graphs also have more variation in their degree centrality than just average degree. This is due to the normalization that occurs for degree centrality. AGV, GEO, LPA, and SMW graphs display very minimal variation in degree centrality across their 1000 graphs.

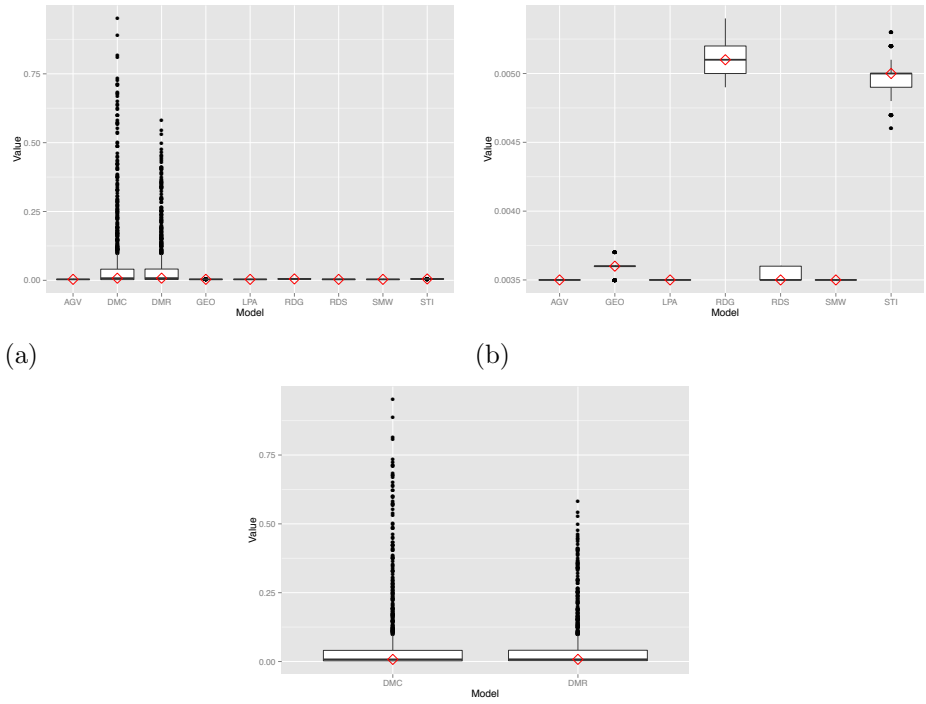


Figure 3.15. **Comparison of the average degree centrality across model graph types.** Each box plot shows the distribution of the average degree centrality for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding DMC and DMR. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of DMC and DMR model graphs.

Eigenvector

Eigenvector centrality, which indicates the influence of a node based on its connections (Newman, 2006), shows much variation across many of the model types (Figure 3.16). Once again, DMC and DMR graphs show the greatest variation making it difficult to infer anything about the other model types (Figure 3.16a). When they are removed we see SMW and AGV graphs both show significant variation. SMW graphs have significant variation in their first quartile. AGV graphs show a similar, but less pronounced, trend. GEO is the opposite, having numerous outliers in the the fourth quartile. The remaining models do not display significant variation (Figure 3.16b). Finally, DMC has a larger IQR than DMR, but DMR has more variation, and more outliers, in its upper quartile (Figure 3.16c).

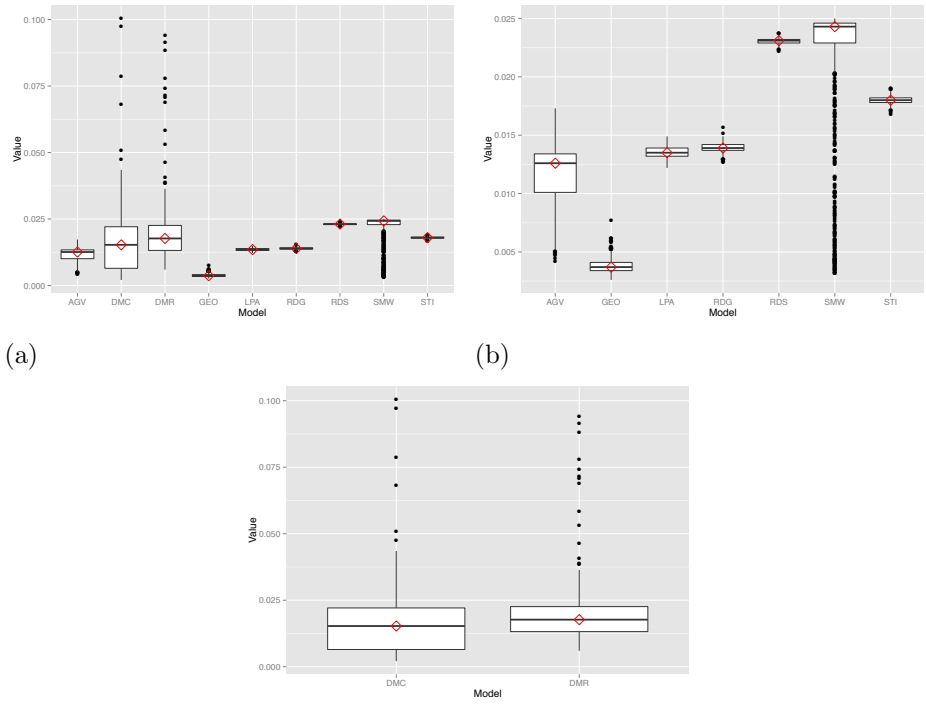


Figure 3.16. **Comparison of the average eigenvector centrality across model graph types.** Each box plot shows the distribution of the average eigenvector centrality for the 1000 graphs a given model type. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Comparison of all model graph types. **(b):** Comparison of model graph types excluding DMC and DMR. These models were removed because their variation eclipsed the variation of the other model types. **(c):** Comparison of DMC and DMR model graphs.

3.4 Discussion

In this chapter we analyzed the variation in graph measures across the 1000 graphs generated by each growth mechanism. A total of fifteen graph measures were considered. Results were displayed using box plots that indicate median and IQR, as well as information about outliers. Median statistics were used due to the variation in measure values presented by some of the model types. The variations produced distributions of values that were not normal, thus not adequately summarized by the mean.

It is very clear that two growth mechanism have the ability to produce very different graphs with each simulation. These are the DMC and DMR algorithms. Of the fifteen measures considered, DMC had ten measures where the variation was so large that it eclipsed the results of all of the other model types, except DMR. DMR did this on eight

occasions. The large variation occurs because the algorithms' treatment of edges. Unlike all of the other model types, DMC and DMR do not take the desired number of edges as an input when the graph is being created. Instead, they add nodes one at a time to the graph creating new edges based on a probability between zero and one. In the event that the probability is high, without any cap to the number of edges, the graphs can grow seemingly unlimitedly. On the other hand, if the probability of connection is low, the graphs can be produced with very few edges. Graphs with few edges tend to have copious amounts of lone nodes, which are removed as directed by the literature (Przulj *et al.* , 2004; Middendorf *et al.* , 2005; Pržulj & Higham, 2006; Przulj, 2007; Su *et al.* , 2011).

While the DMC and DMR graphs could be moderated by using the desired number of edges as an input value, similar to RDG, we choose to not follow that path for the remainder of this dissertation for two reasons. First, the literature used these model graphs built with the same algorithm (Middendorf *et al.* , 2005; Su *et al.* , 2011). In order to present results that are fully comparable, we must use the exact same model types. Secondly, these model types were designed in conjunction with biologists to specifically mimic the way in which protein-protein interaction networks are thought to be built. Causing significant changes to the structure of their growth mechanisms may result in the loss of the biological inference that went into their creation.

DMC and DMR graphs were not the only model types to present a significant amount of variation across their 1000 graphs. AGV and SMW both showed a significant amount of variation. The variation in AGV graphs occurred largely when paths were examined, such as in diameter and radius, and in the centrality measures. We speculate that this variation occurs because AGV graphs are growing and through that process they have the possibility to evolve scale-free properties. In the instances where these properties evolve, the distance between any two nodes becomes ultra-small (Watts & Strogatz, 1998). Since this is not a guaranteed feature, we see lots of variation in related measures.

The variation in the SMW graphs is particularly interesting because it occurs during many of the same measures as AGV. This is odd because the idea behind SMW graphs is that every node is available to every other node within a small number of steps, usually about six (Watts & Strogatz, 1998). The large variation in the radius and diameter of SMW

graphs indicates that the growth mechanism does not always succeed in making graphs with the properties that it intends to.

Another instance of the growth mechanism not building models with the intended features is displayed by LPA graphs. These model graphs are built to display the scale-free properties defined by Barabasi (Barabási & Albert, 1999; Albert & Barabási, 2002). Therefore, we expect that they should have a larger S-metric value than all the other model graph types. Instead, LPA graphs had one of the lowest median values for the S-metric. Low S-metric values indicate a scale-rich graph, as opposed to a scale-free. The inconsistency displayed by this growth mechanism can be attributed to variations in the definition of a scale-free network. This type of network is often defined simply by its power-law degree distribution, which is most often the result of using linear preferential attachment to build the graph (Barabási & Albert, 1999; Barabási *et al.* , 2000; Albert & Barabási, 2002; Li *et al.* , 2005). The creators of the S-metric, however, have a more rigorous definition (Li *et al.* , 2005). This definition requires features that are often not present when the graph is built using linear preferential attachment. Such features include a hub-like core and self-similarity. Therefore, we can infer that the LPA graphs created here do not fit the latter criteria of the scale-free network, but they may fit the less rigorous definition proposed by Barabasi (Barabási & Albert, 1999; Albert & Barabási, 2002).

The models with the least variation are LPA and RDS. Neither of these graphs showed any significant variation across their 1000 model graphs in comparison to the variation seen by the other model types. This indicates that these model types are very consistent in the structure of graph that they build. GEO, RDG, and STI all showed minimal variation. The latter two model types, RDG and STI, both have growth mechanisms that are prone to the creation of lone nodes (that are then eliminated from the graph), thus most of their variation occurred in the numbers of nodes and edges.

Graphs created using the GEO growth mechanism showed minimal amounts of variations for most of the measures. The amounts increased for the diameter, radius, and average shortest path length. It is interesting to note that for these three measures, GEO reported higher values than the all other model types.

Looking across all of the measure, we see that median values for most of them are very similar across model types. This implies that they are all recreating features seen in the *S. cerevisiae* PPI network that was used as the basis for their creation. Results for those comparisons are seen in Chapter 4.

Finally, purpose of this chapter was to determine the variation in features of graphs built using the same growth mechanism. The underlying question behind this is whether two graphs built with the same growth mechanism should automatically be classified as the same model type. This is an especially important question when considering the great variation posed by the DMC and DMR graphs. Our answer: not necessarily. For the remainder of this dissertation, however, we will continue to treat graphs created by the same growth mechanism as representative of the same model type in order to have an accurate comparison to the current literature. In Chapter 14, we discuss the steps that can be taken when working with growth mechanisms that result in graphs with a large amount of variation.

Chapter 4

Measure Based Comparison of Model Graphs v *Saccharomyces cerevisiae* PPI Network

Researchers have attempted to classify PPI networks in numerous ways, often designing new metrics in the process. No researcher, however, has compared a real-world PPI network to multiple model graphs based simply on an all-encompassing array of network measures. This is the aim of this chapter. A total of eighteen measures are considered; some are graph-level measures while others report median values of node-level measures.

4.1 Methods

The creation of the model graphs was based on features of the *S. cerevisiae* PPI network. Despite this, the different models gave rise to graphical features that differed drastically from the empirical network and each other. For the 1000 graphs of each type, the median value of each metric was calculated. Medians were used, as opposed to means, due to their robustness to extreme values. This was necessary because the DMC and DMR models were so varied in several of their feature values that their skewed results made average values unreliable. The graph measures are evaluated in two ways. The first is for statistical significance. We are looking for evidence that the median for each measure is equal to the *S. cerevisiae* PPI network value. Our hypotheses are as follows:

$$H_0 : \text{model median} = \text{empirical value} \quad H_A : \text{model median} \neq \text{empirical value} \tag{4.1}$$

Since we know that the majority of our values do not follow the normal distribution (e.g. Figure 4.3) we are restricted to non-parametric tests. Many of the common non-parametric tests, such as the Wilcoxon signed-rank test, require symmetry around the median. Since we can also not make such an assumption, especially since some of our measures such as number of nodes have caps on their size, we use a slightly less powerful test, the Wilcoxon Sign test. The Wilcoxon Sign test makes three assumptions.

1. The observations are independent.

2. The observations come from the same population.

3. The observations are ordered so that comparisons "greater than", "less than", and "equal to" carry meaning.

Since these assumptions are met, we evaluate the statistical significance at the 95% significance level with this test. We are looking for model types whose measures accept the null hypothesis.

The other method of evaluating the model graphs is slightly less rigorous. Since the problem of network classification exists, we can infer that it is not easy to correctly match all of the desired features. Therefore, in addition to a lack of statistically significant difference, we identify measure values that fall within ±5% of the empirical value. Models with values within this range are considered to have successfully replicated the specific measure. The total number of successful replications, or matches, is calculated to obtain an overall impression of each model types' ability to mimic features. The category of model graph with the largest number of matches can be considered the best structural fit out of the considered types.

4.2 Results

The networks measures mentioned in Section 1.2 can be divided into four descriptive categories: size, distance, centrality, and connection. Results are shown in Tables 4.1 - 4.8. In these tables, values that are not within ±5% of the empirical value of shown in bold and statistically significant values are marked with an asterisk (*). Therefore a value that matches under both analyses has no designations.

4.2.1 Size Measures

The first category, size measures, consists of the number of nodes (n), number of edges (m), proportion of nodes in the giant component ($|\mathcal{V}_{\mathcal{H}}| \,/\, |\mathcal{V}_{\mathcal{G}}|$), and density ($\mathcal{D}$) of the graphs, Table 4.1.

In Figure 4.1, we can see how the model graphs (in blue) compare to the empirical network (in red) overall. From this image, it is clear that the many of the model graphs do a good job recreating desired size features. There are at least two models though, DMC and

DMR, that are very different from the others. They differ dramatically in size, alternating between smaller numbers of nodes, larger numbers of edges, and extremely small proportion of nodes present in the giant component. Two others, RDG and STI, have fewer nodes but fall back into line with the majority of the models for the other features.

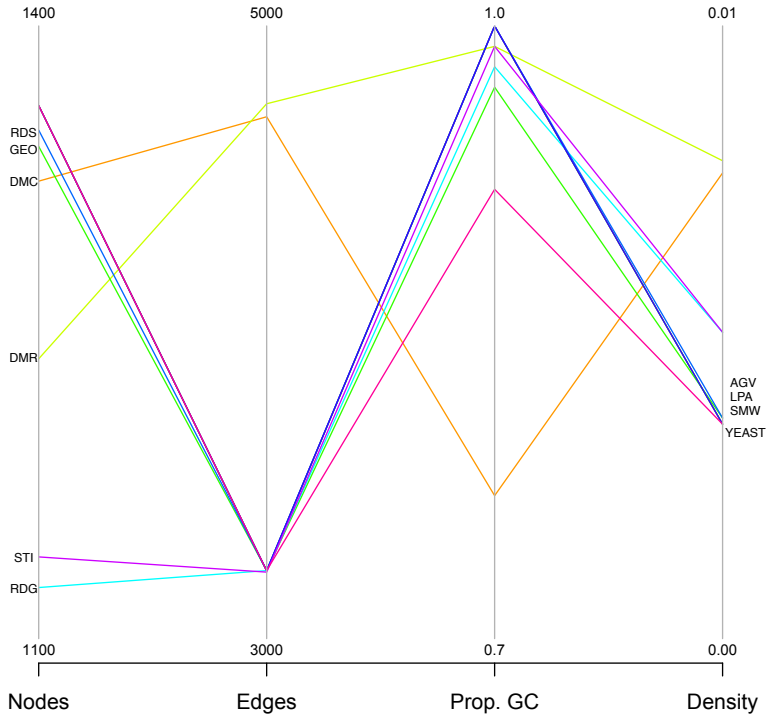


Figure 4.1. **Parallel coordinate representation of size measures.** Measures included are the number of nodes (Nodes), number of edges (Edges), proportion of nodes in the giant component (Prop. GC), and density. All of these measures are graph-level, thus the values represented are the medians of the 1000 simulated graphs for each of the nine model types.

Looking more deeply into the size features, we notice several interesting things. Despite all of the model graphs being created with the intention of having 1361 nodes just like the *S. cerevisiae* PPI network, the majority of the graphs did not produce this as their median value, Table 4.1. Three model types do not produce any lone nodes: AGV, LPA, and SMW. In every situation, these graphs have 1361 nodes. Interestingly, these model types also only build a graph with one component containing all of the nodes. The model types most inclined to produce lone nodes are RDG and STI. It is known, however, that DMC and DMR networks produce the smallest graphs of the set with respect to number of nodes, Table 4.2.

Table 4.1. Median values of simulated model graph size measures.

	n	m	$\lvert\mathcal{V}_{\mathcal{H}}\rvert \,/\, \lvert\mathcal{V}_{\mathcal{G}}\rvert$	$\mathcal{D}$
Yeast Data	1361	3222	0.92	0.0035
AGV	1361	3221*	**1.0***	0.0035
DMC	1324*	**4703.5***	**0.77***	**0.0076***
DMR	**1237***	**4746***	**0.99***	**0.0078***
GEO	1341	3222	0.97*	0.0036*
LPA	1361	3222	**1.0***	0.0035
RDG	**1125***	3222	**0.98***	**0.005***
RDS	1349*	3222	**1.0***	0.0036*
SMW	1361	3222	**1.0***	0.0035
STI	**1140***	3217*	**0.99***	**0.005***

Table shows the values of the size measures. The size measures included in the table are the number of nodes (n), number of edges (m), proportion of nodes in the giant component ($\lvert\mathcal{V}_{\mathcal{H}}\rvert \,/\, \lvert\mathcal{V}_{\mathcal{G}}\rvert$), and density ($\mathcal{D}$). Each value is the median across the 1000 model graphs of the given type. Values written in boldface are not within $\pm 5\%$ of the empirical value. Values with an asterick (*) are statistically significantly different than the empirical value.

The distribution of the number of nodes for each of the models type can be seen more precisely in Figures 4.2, 4.3. The distribution of AGV, LPA, and SMW is not shown in either histogram because these model types only produce graphs of one size. In addition, it is necessary to show DMC and DMR histograms on a different set of axes because their range is far more spread out than the range of GEO, RDG, RDS, and STI. In Figure 4.2, we see that the range of numbers of nodes goes from 1100 to 1361. GEO and RDS are both clustered toward the top end of the range while the other two networks, RDG and STI are clustered near the bottom. All of the distributions do appear normal. In Figure, 4.3 the

Table 4.2. Ranges of model graph size based on numbers of nodes and edges.

	# Nodes			# Edges		
	Min	IQR	Max	Min	IQR	Max
AGV	1361	0	1361	3216	2	3222
DMC	17	276.75	1361	91	34674.25	880237
DMR	31	372	1361	21	35712.75	537036
GEO	1324	6	1353	3222	0	3222
LPA	1361	0	1361	3222	0	3222
RDG	1093	14	1151	3222	0	3222
RDS	1339	4	1358	3222	0	3222
SMW	1361	0	1361	3221	0	3222
STI	1092	17	1187	3057	73.25	3408

Table gives the minimum, maximum, and IQR for the number of nodes and the number of edges. The IQR is a measure of spread. It give the difference between the first and third quartile values.

distributions are far from normal, instead they display a negative skew with a long left tail. The peak of the number of nodes is at 1361 for both networks, but in this instance the range begins just above zero. It is interesting to note that while RDG and STI consistently produce lone nodes, DMC and DMR only do so occasionally, but their results are far more drastic when it does happen.

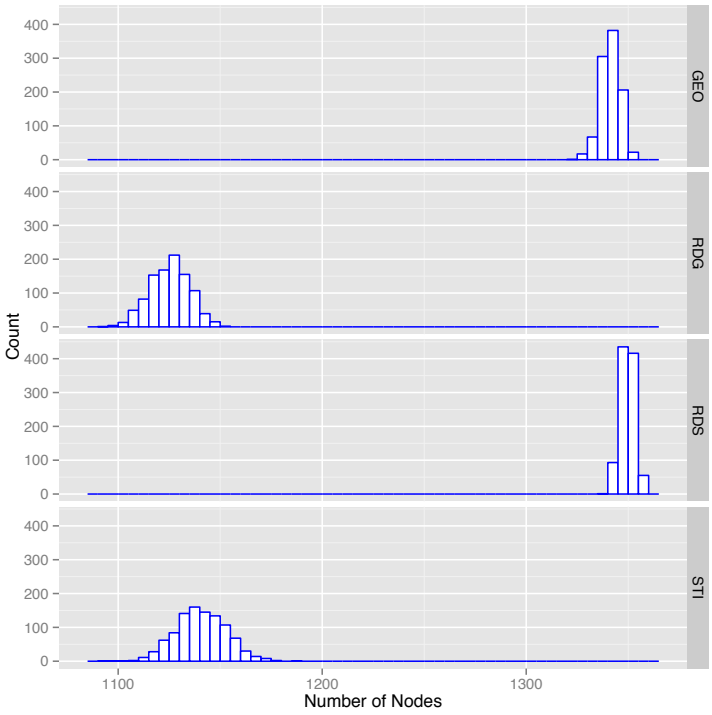


Figure 4.2. **Histograms of number of nodes for GEO, RDG, RDS, STI model graphs.**

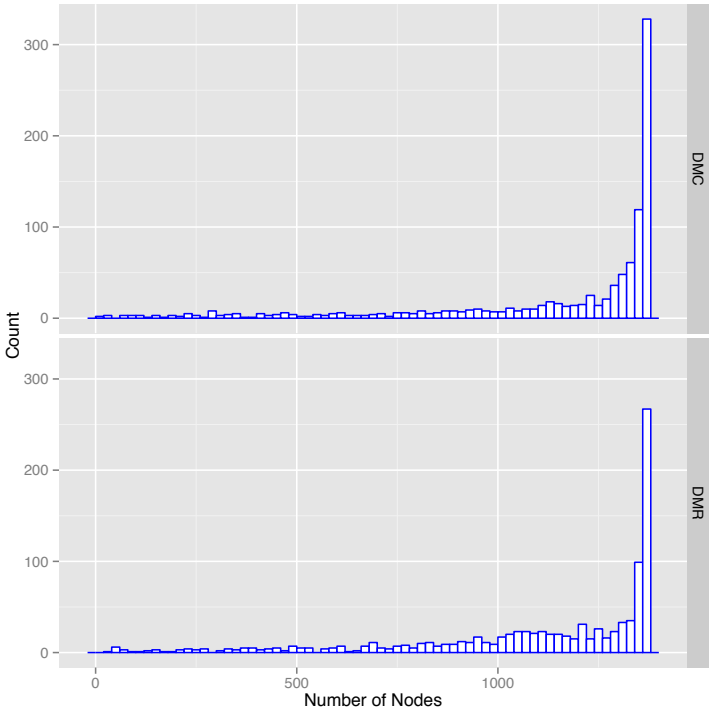


Figure 4.3. **Histograms of number of nodes for DMC, DMR model graphs.**

The majority of graphs constructed had the same number of edges as the real-world network, with seven model types having median values of 3222. The only two model types that did not were DMC and DMR, both of which had a median of nearly 1500 more nodes (Table 4.2). Similar numbers of nodes and edges in the model graphs had similar density values across the board with the only differences coming from the DMC and DMR.

One of the more challenging attributes for the model graphs to mimic is the proportion of nodes in the giant component. Though the majority of models do peak similarly to the *S. cerevisiae* PPI network in Figure 4.1, only one of these values is within 5% of the proportion of nodes in the PPI network giant component and none of them fail to be statistically significantly different, Table 4.1. This model graph is GEO. The graph model that failed most impressively to mimic this feature was DMC, whose proportion was 0.7713 compared to *S. cerevisiae* PPI network's proportion of 0.9155.

4.2.2 Distance Measures

The next category of measures examines the distance between nodes in a graph. These distances can be measured in multiple ways (e.g. diameter, radius, and average shortest path length), but they can only be calculated on the giant component of a graph. If we look at the big picture in Figure 4.4, only one model graph has truly substantial deviation from the empirical distance values. It is also interesting to note that there are more occurrences of matching radius values than either diameter or average shortest path length.

Now looking at the distance measure values in more detail (Table 4.3), it is clear that no model graph has the same diameter as the yeast data, however RDG is the closest with 11 v 12. Two network models, RDS and RDG, have the same radius. Five more, AGV, DMC, DMR, LPA, and STI, are all very close, 5 v 6. Two models, AGV and RDS, have a similar average shortest path length.

Table 4.3. Median values of simulated model graph distance measures.

	$diam(\mathcal{G})$	$rad(\mathcal{G})$	$\bar{\ell}$	SMW	SF
Yeast Data	12	6	4.90	Y	N
AGV	**8***	**5***	4.67*	Y	N
DMC	**9***	**5***	**3.62***	**N**	N
DMR	**9***	**5***	**3.72***	**N**	N
GEO	**34**	**19***	**13.75***	**N**	N
LPA	**7***	**5***	**4.23***	Y	N
RDG	11*	6	**4.27***	**N**	N
RDS	**10***	6	4.80*	**N**	N
SMW	**9***	**7***	**5.15***	Y	N
STI	**9***	**5***	**3.88***	Y	N

Table shows the values of the distance measures. The distance measures included in the table are the diameter ($diam(\mathcal{G})$), radius ($rad(\mathcal{G})$), average shortest path length ($\bar{\ell}$), small-world property (SMW), and scale-free property (SF). Each value is the median across the 1000 model graphs of the given type. Values written in boldface are not within ±5% of the empirical value. Values with an asterick (*) are statistically significantly different than the empirical value. For the SMW and SF columns, a Y indicates that the based on the most lenient criteria applied to the median value (Table 4.4), the model type has the property. An N indicates that it does not. The most lenient criteria are: $p \gg 1$ if and only if $p > 3$ and $q \approx 1$ if and only if $q \in [0.85, 1.15]$.

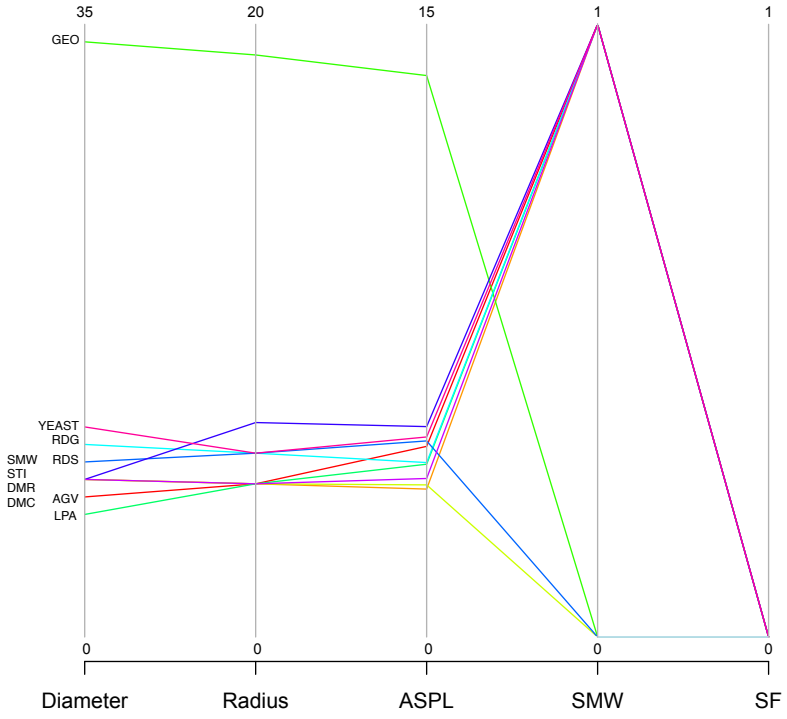


Figure 4.4. **Parallel coordinate representation of distance measures.** Measures included are the diamerter, radius, average shortest path length (ASPL), small-world property (SMW), and scale-free property (SF). The latter two properties are binary where zero means the model type does not possess the given property and one means that it does. All of these measures are graph-level, thus the values represented are the medians of the 1000 simulated graphs for each of the nine model types.

The last two columns in Table 4.3 refer to the small-world and scale-free properties from Section 1.2. Recall that, we can say a graph has small-world properties if:

$$p = \frac{\bar{C}}{\bar{C}_r} \gg 1 \tag{4.2}$$

$$q = \frac{\bar{\ell}}{\bar{\ell}_r} \approx 1, \tag{4.3}$$

and a graph has scale-free properties if:

$$s = \frac{\bar{\ell}}{\bar{\ell}_r} \approx 1. \tag{4.4}$$

Median calculation for p, q, and s are seen in Table 4.4. Values in bold indicate that the values obviously do not meet the listed requirement. None of the s were approximately equal to one, even with a generous definite of approximate, and only one was larger than one (GEO). Values for p and q more consistently achieve the value required for the property. In Table 4.3, models with the small-world or scale-free property are designated with a Y, those without the property have an N. The only model types that do demonstrate the small-world property found in the *S. cerevisiae* PPI network are AGV, LPA, SMW, and STI. None of the models, including the PPI network, display the scale-free property. This is interesting considering that the LPA was designed with this feature in mind.

If we examine the exact values for each of the 1000 model graphs, as opposed to using the median values in Table 4.4, we can see the exact percent of graphs that possess the SMW and SF properties (Table 4.6). Since we are examining each number individually it is necessary to make concrete rules that dictate whether a value is much greater than one and whether a value is approximately equal to one. Therefore we propose three schemes with varying levels of stringency (Table 4.5). In Scheme A, a value much be larger than 10 in order to be considered much greater than one (Heath *et al.* , 1956) and a value much be between 0.95 and 1.05 in order to be considered approximately equal to one. In Scheme B, the criteria loosens to include values greater than 5 and between 0.90 and 1.10 for the two criteria respectively. In the final scheme, C, the restrictions loosen even more. Now values

Table 4.4. Calculations to determine small-world and scale-free properties

	$p \gg 1$	$q \approx 1$	$s \approx 1$
Yeast Data	62.38	1.05	0.78
AGV	23.11 (57.90)	1.01 (0.18)	**0.74 (0.13)**
DMC	15.76 (49.03)	**1.34 (0.52)**	**0.58 (0.64)**
DMR	**0.99 (1.02)**	1.15 (0.35)	**0.59 (0.53)**
GEO	135.2 (3.72)	**3.00 (0.15)**	**2.18 (0.11)**
LPA	3.28 (1.94)	0.91 (0.05)	**0.67 (0.03)**
RDG	**2.18 (0.38)**	1.06 (0.01)	**0.68 (0.008)**
RDS	**0.91 (0.34)**	1.04 (0.003)	**0.76 (0.003)**
SMW	21.92 (64.52)	1.11 (0.19)	**0.82 (0.14)**
STI	3.44 (0.63)	0.95 (0.008)	**0.62 (0.007)**

Table shows the median p, q, and s across the 1000 model graphs of each type. Values in boldface mean that the value obviously not meet the requirements: $p \gg 1$, $q \approx 1$, and $s \approx 1$. The values in parentheses are the IQR.

greater than 3 are considered much greater than one and values between 0.85 and 1.15 are approximately equal to one.

Table 4.5. Schemes to determine if the small-world and scale-free properties were met.

	$p \gg 1$	$q \approx 1$	$s \approx 1$
Scheme A	$p > 10$	$q \in [0.95, 1.05]$	$s \in [0.95, 1.05]$
Scheme B	$p > 5$	$q \in [0.90, 1.10]$	$s \in [0.9, 1.1]$
Scheme C	$p > 3$	$q \in [0.85, 1.15]$	$s \in [0.85, 1.15]$

The definitions to determine whether the small-world and scale-free properties have been met are vague. This table presents three different definitions for what being much greater than one or approximately equal to one means. These different definitions are presented as schemes A, B, and C.

Under Scheme A, no model graph has the scale free property (Table 4.6). A total of 28.6% of the AGV graphs have the SMW property as do 3.2% of DMC graphs. Upon moving to Scheme B, these values increase to 52.1% and 5.7% respectively. In addition, 18.8% of SMW graphs now display the SMW property. Also in Scheme B do we see a few graph displaying the SF property. Most of these are DMC, with a total of 0.6% of its graphs showing SF properties. Finally, in the most lenient scheme STI graphs have the highest percentage displaying SMW property (83.4%). This sudden increase from Scheme B to Scheme C implies that many of its p and q values must be borderline. We do not see an

equally impressive jump for SF properties indicating that most s values are not borderline.

Table 4.6. Percent of model graphs that have the small-world or scale-free property stratified by model type.

	Scheme A		Scheme B		Scheme C	
	SMW	SF	SMW	SF	SMW	SF
AGV	28.6	0.0	52.1	0.1	75.9	0.3
DMC	3.2	0.0	5.7	0.6	9.6	2.4
DMR	0.0	0.0	0.0	0.0	0.2	0.0
GEO	0.0	0.0	0.0	0.0	0.0	0.0
LPA	0.0	0.0	0.2	0.0	48.8	0.0
RDG	0.0	0.0	0.0	0.0	0.4	0.0
RDS	0.0	0.0	0.0	0.0	0.0	0.0
SMW	0.0	0.0	18.8	0.4	36.9	1.0
STI	0.0	0.0	0.1	0.0	83.4	0.0

Table shows the exact percent of the 1000 graphs of each model type that display SMW or SF characteristics. There is some ambiguity regarding whether a value is much greater than one or approximately equal to one. Therefore, we use three different schemes with different levels of rigor. In scheme A, $p \gg 1$ if and only if $p > 10$ and $q \approx 1$ if and only if $q \in [0.95, 1.05]$. In scheme B, $p \gg 1$ if and only if $p > 5$ and $q \approx 1$ if and only if $q \in [0.9, 1.1]$. In scheme C, $p \gg 1$ if and only if $p > 3$ and $q \approx 1$ if and only if $q \in [0.85, 1.15]$.

4.2.3 Centrality Measures

Centrality measures in this paper were averaged over all the nodes in each network. Averages of each graph's centrality values were calculated at this stage, as opposed to median values, to conform with existing literature. It is more common to see an average centrality expressed, less common to see a median value. Medians of the 1000 model graphs' average values were then calculated. Medians were used over means due to the large variation in size and shape seen by the models, specifically the biologically motivated DMC and DMR networks.

Overall, in the centrality measures we see numerous networks mimicking the pattern displayed by the *S. cerevisiae* PPI network, simply transposed higher on the graph, Figure 4.5. Only two of the centrality measures, degree (DC) and betweenness (BC), are replicated with similar values. The replication of the two other centralities, closeness (CC) and eigenvector (ψ), resulted most often in larger values.

As displayed by Figure 4.5, Table 4.7 shows there are not many models whose centrality values are within 5% of the yeast centrality values. The only model that has

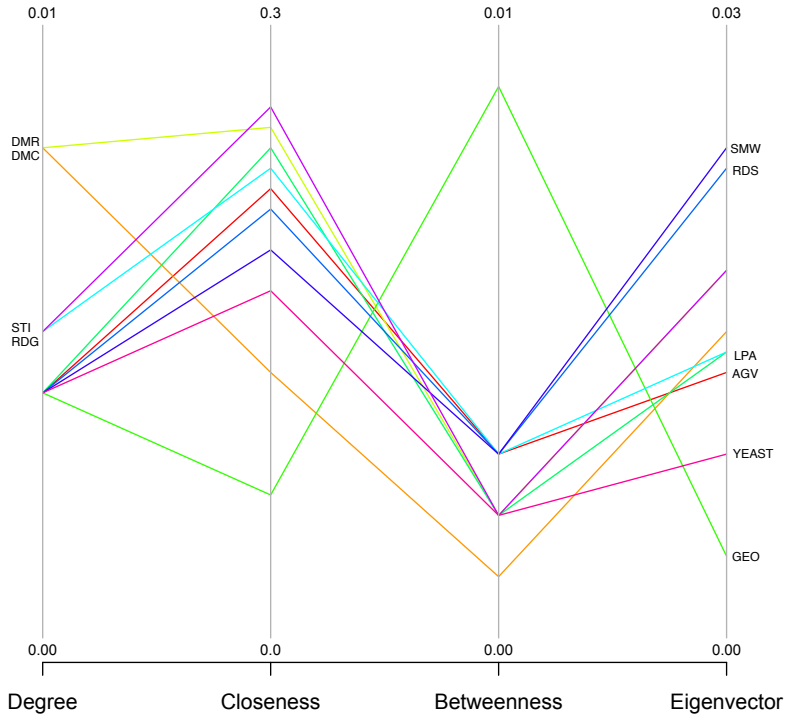


Figure 4.5. **Parallel coordinate representation of centrality measures.** Centralities included are degree, closeness, betweenness, and eigenvector. Values are calculated from the 1000 simulated graphs for each of the nine model types. Since centrality measures are node-level properties, the average centrality for each graph is calculated and then the median value is taken from those 1000 averages. This median is the value presented.

Table 4.7. Median values of simulated model graph centrality measures.

	DC	CC	BC	ψ
Yeast Data	0.004	0.17	0.002	0.009
AGV	0.004	**0.22***	**0.003***	**0.013***
DMC	**0.008***	**0.13***	**0.001***	**0.015***
DMR	**0.008***	**0.25***	0.002	**0.018***
GEO	0.004	**0.07***	**0.009***	**0.004***
LPA	0.004	**0.24***	0.002	**0.014***
RDG	**0.005***	**0.23***	**0.003***	**0.014***
RDS	0.004	**0.21***	**0.003***	**0.023***
SMW	0.004	**0.19***	**0.003***	**0.024***
STI	**0.005***	**0.26***	**0.002***	**0.018***

Table shows the values of the centrality measures. The centrality measures included in the table are degree (DC), closeness (CC), betweenness (BC), and eigenvector (ψ). Since these are node-level measures, the average is taken within a graph to obtain a since value. Then the median is taken across the 1000 model graphs. This is the value presented.Values written in boldface are not within $\pm 5\%$ of the empirical value. Values with an asterick (*) are statistically significantly different than the empirical value.

more than one match is LPA, which matches on both degree and betweenness centrality. There are no matches for eigenvector centrality and only one for closeness centrality implying that these two features are more difficult to mimic.

4.2.4 Connection Measures

The final category of measures considered is connection. There are five connection measures considered: average degree ($\bar{k}$), S-metric ($\mathcal{S}(\mathcal{G})$), assortativity ($r(\mathcal{G})$), transitivity or global clustering coefficient ($C(\mathcal{G})$), and average clustering coefficient ($\bar{C}$). It is important to note that just like the centrality measures, the values listed for the average degree and average clustering coefficient are the medians of the 1000 averages within each model graph types. This was done, once again, to preserve consistency across literature. The parallel coordinate plot for connection measures is significantly different than the previous plots in that no underlying order is present, Figure 4.6. Average degree and average clustering coefficient are the only measures replicated similarly in the model graphs, the others are all vastly different in patternless ways.

An examination of the specific connection measure values supports the ideas ascertained from Figure 4.6. Average degree was the simplest connection measure to replicate. Five of the model graphs had values within 5% of the empirical value. Only one of the

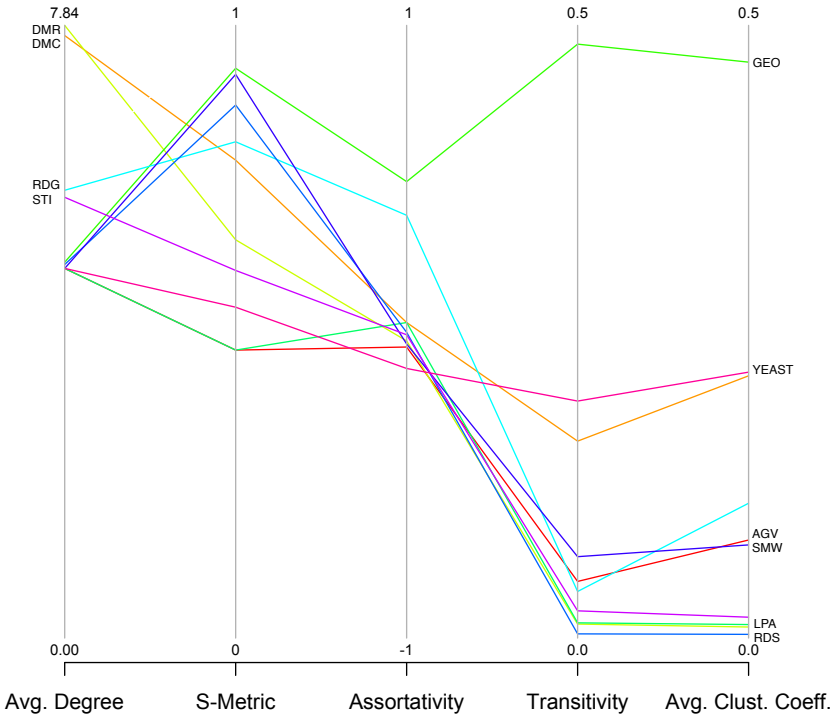


Figure 4.6. **Parallel coordinate representation of connection measures.** Measures included are the average degree (Avg. Degree), S-metric, assortativity, transitivity, and average clustering coefficient (Avg. Clust. Coeff). Average degree and average clustering coefficient are node-level measure, thus the average value for each graph is calculated and then the median value is taken from those 1000 averages. This median is the value presented. The remaining measures, S-metric, assortativity, and transitivity are all graph-level, thus the values represented are the medians of the 1000 simulated graphs for each of the nine model types.

values of the average clustering coefficient values was a match, DMC. None of the other measures were replicated with any measure of accuracy (Table 4.8).

Considering the S-Metric, which determines the amount of hub-like behavior in a network, conflicting results were obtained. The two networks that theoretically should have high values for the S-Metric, LPA and AGV, exhibit the two lowest values.

As previously mentioned, most biological networks display negative assortativity, or dissortativity, meaning that low-degree nodes are connected to high degree nodes. The graph types most often described as the best model for PPI networks (DMC, GEO, LPA, RDG) are the only ones not showing signs of dissortativity. Their assortativity coefficients are all positive. RDS has an assortativity coefficient that is close to zero and thus it has neither assortative nor dissasortative characteristics.

Just like the previous two measures, none of the model graphs have a transitivity value within 5% of the observed value. Only one graph has a larger transitivity, GEO, and this value is approximately 2.5 times larger than the empirical value. The remaining value are all smaller, ranging from 27% to 98% less than the empirical value.

Table 4.8. Median values of simulated model graph connection measures.

	$\bar{k}$	$S(\mathcal{G})$	$r(\mathcal{G})$	$C(\mathcal{G})$	$\bar{C}$
Yeast Data	4.73	0.54	−0.12	0.1934	0.217
AGV	4.73	**0.47***	**-0.05***	**0.05***	**0.08***
DMC	**7.71***	**0.78***	**0.03***	**0.16***	0.214
DMR	**7.84***	**0.65***	**-0.03***	**0.01***	**0.009***
GEO	4.81	**0.93**	**0.49**	**0.48***	**0.47***
LPA	4.73	**0.47***	**0.03***	**0.01***	**0.011***
RDG	**5.73***	**0.81***	**0.38***	**0.04***	**0.11***
RDS	4.78*	**0.87***	**-0.002***	**0.003***	**0.003***
SMW	4.73	**0.92***	**-0.04***	**0.07***	**0.076***
STI	**5.64***	**0.60***	**-0.01***	**0.02***	**0.017***

Table shows the values of the connection measures. The connection measures included in the table are the average degree ($\bar{k}$), S-metric ($S(\mathcal{G})$), assortativity ($r(\mathcal{G})$), transitivity ($C(\mathcal{G})$), and average clustering coefficient ($\bar{C}$). For the graph-level measures (S-metric, assortativity, and transitivity), each value is the median across the 1000 model graphs of the given type. The remaining measures are node-level. For these, the average is taken within a graph to obtain a since value. Then the median is taken across the 1000 model graphs. This is the value presented. Values written in boldface are not within ±5% of the empirical value. Values with an asterick (*) are statistically significantly different than the empirical value.

4.2.5 Biologically Significant Measures

A final way to consider the model graphs ability to replicate essential features of the *S. cerevisiae* PPI network is through a comparison of the biologically significant features with respect to PPI networks. These features are density, transitive, average degree, and assortativity. From this list it is obvious that how nodes are connected to each other, and how many of these connections exist, is an essential part of biological networks. In Figure 4.7, a bit of a reciprocal relationship is evident. Those graphs with high density often have low transitivity, high average degree, and low assortativity and vice versa. Several model graphs do follow the same general shape as the empirical network, but these are no close matches for transitivity or assortativity just as previously mentioned.

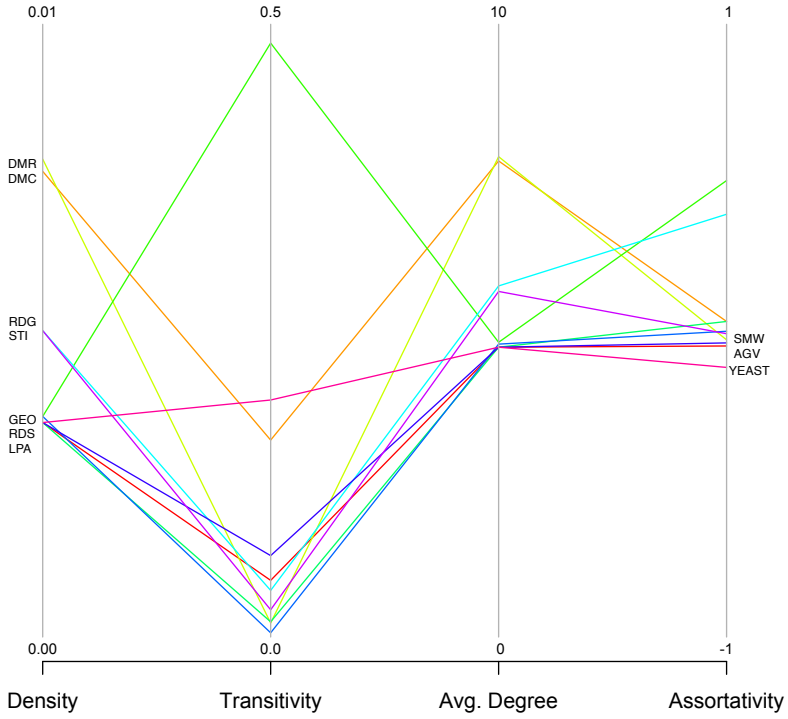


Figure 4.7. **Parallel coordinate representation of biologically significant measures.** Measures included are the density, transitivity, average degree (Avg. Degree), and assortativity. Average degree is a node-level measure, thus the average value for each graph is calculated and then the median value is taken from those 1000 averages. This median is the value presented. The remaining measures, density, transitivity, and assortativity are all graph-level, thus the values represented are the medians of the 1000 simulated graphs for each of the nine model types.

4.2.6 Summary of Measure Based Comparison Broken Down by Category

The number of model graphs whose median values were not statistically significantly different at the 95% significance level are shown in the first column of Table 4.9. Values within ±5% of the *S. cerevisiae* PPI network values is shown in the second column. Three models are tied as having the highest number of values reproduced within 5% of the median values, AGV, LPA, and RDS. Interestingly, all of LPA's values that are within 5% are also statistically significant. This is not the case for AGV and RDS (6 v 4). Therefore, we can conclude that the features LPA successfully mimicked it did so almost perfectly. The other models were close in their recreations, but still not quite as accurate.

Table 4.9. Model graph and *S. cerevisiae* PPI network matches based on graph measures.

	Statistical Matches	±5% Matches
AGV	6	8
DMC	3	4
DMR	2	2
GEO	4	7
LPA	8	8
RDG	4	5
RDS	4	8
SMW	7	7
STI	2	3

This table shows the number of times each model is not statistically significant different from the empirical value (Statistical Matches) and the number of times each is within ±5% of the empirical value (±5% Matches). It is summarized across the four categories discussed in Sections 4.2.5 to 4.2.1.

The model that performed the worst is DMR, which scored only a two under both methods of analysis. This was the model that Su *et al.* found to be as their best fit for the PPI networks (Su *et al.* , 2011). Przulj found the best fit to be the GEO (Przulj *et al.* , 2004; Przulj, 2007) or STI (Pržulj & Higham, 2006) network depending on the method and model choices utilized. While GEO scored a respectable seven on the second analysis, only four of its measures were not statistically different from the ideal value. STI did barely better than DMR in its ability to match measure values with two values not statistically different and three values within 5%. It should be noted that while LPA scored the highest number of matches, it still matched only 44% of the metrics correctly.

Table 4.10. Model graph and S. cerevisiae PPI network matches based on graph measures, stratified by measure category.

	Size (4)	Distance (5)	Centrality (4)	Connection (5)
AGV	3	3	1	1
DMC	1	2	0	1
DMR	0	1	1	0
GEO	4	1	1	1
LPA	3	2	2	1
RDG	1	4	0	0
RDS	3	2	1	1
SMW	3	2	1	1
STI	1	2	0	0
% Matched	**53**	**22**	**19**	**13**

This table only shows matches based on the number of times each model is within ±5% of the empirical value. Matches are stratified by measure category. The number is parentheses under the measure category indicates the number of measures in that category.

Stratifying the results by measure category shows that categories were not all matched at the same rate. The results in Table 4.10 show the stratification only for values within 5% of the empirical value because there were significantly more matches in that category (32% v 25%) . The majority of the matches occurred in size measures with 53% of values properly replicated. Only 13% of the connection measures matched within the designated range. Distance and centrality measures were both replicated 22% and 19% of the time respectively.

4.3 Discussion

In this chapter, the *S. cerevisiae* PPI network was compared to the nine model graphs using eighteen different network measures. Model graphs were considered a match for the empirical network under two circumstances. First, if the median value for the model graph was within 5% of the empirical value, it was considered a match. Second, if the median value for the model graph was not statistically significantly different than the empirical value, it was considered a match. The second method utilized the Wilcoxon Sign test to quantify statistical significance.

The differences in network size, judged either on number of nodes or number of edges, are direct results of the way the algorithms build the graphs. This is mentioned

in Chapter 3. The failure to perfectly replicate the desired number of nodes is due to 1) the allowance of lone nodes in the model building algorithms and their subsequent lack of inclusion; and 2) the cap on the number of nodes allowed. This cap comes from the use of the number of nodes as input for model creation. Therefore, it is possible for a model graph to have fewer nodes than requested, but impossible for it to have more as the number was capped at 1361. This is important to remember when considering the distribution of the number of nodes. Model types classified as growing, Table 3.1, are more likely to produce lone nodes and thus have lower medians than static graphs (1258 v 1350).

We reiterate the we choose to continue using the model graphs that do not have the desired number of edges. We justify this choice in two ways. First, literature indicates that eliminating lone nodes is a common way of dealing with them (Su *et al.* , 2011; Middendorf *et al.* , 2005; Przulj *et al.* , 2004; Pržulj & Higham, 2006; Przulj, 2007; Kuchaiev *et al.* , 2011) . Second, in order to assess the reproducibility of the results seen in literature we need to use the exact same model graphs. Modifying a model graph algorithm to ensure the number of nodes in the resulting graph directly equals the number inputted is a major modification to the algorithm. The results would not be directly comparable to previous ones. There are some ramifications to this decision of course. The main issue being that some model types may be disadvantaged at classification time. Conveniently, the classifiers discussed in Chapter 5 all have ways of comparing graphs of grossly different sizes without imposing a penalty. Thus, the inclusion of these graphs should be considered a non-issue,

In terms of number of edges, the only two models that did not produce median values within $\pm 5\%$ of the empirical value were those that did not take number of edges as an input value. For DMC and DMR, connections are made based on a uniformly random sampled value p and removed with uniformly random sampled value q.

One of the more interesting features discovered is that the LPA model graphs do not possess scale-free features. This statement is supported by the s value in Table 4.4 and by the S-metric in Table 4.8. In both situations, LPA is expected to have a value very close to one but empirically it produces one of the values farthest from one (0.67, 0.47). There are several possibilities for this. Since those networks were specifically constructed to have a hub-like core, it can be inferred that they do not exhibit self-similarity, the presence of a

self-repeating pattern (Song *et al.* , 2005). Thus these networks can be considered scale-rich as opposed to scale-free (Li *et al.* , 2005). A second explanation for this inconsistency is that the model growth mechanisms used for this analysis are not large enough to produce a truly scale-free graph.

Overall, the ability of the nine types of model graphs to accurately replicate a variety of network measures leaves much to be desired. There is no easy or obvious best answer. Judging from network measures, or at least the measures examined here, we cannot classify the *S. cerevisiae* PPI network. Thus priorities will have to be listed and compromises made in order to find a best fit, and that best fit may change as priorities change.

Chapter 5

Introduction to Network Classification Methods

The size and overall complexity of the *S. cerevisiae* PPI network requires a more analytical method of comparing networks than a simple analysis based on various metric values. Numerous methods have been proposed for the classification of large networks. These methods typically fall into one of two categories: large-scale, focusing on the larger topological features and graph-level properties; and small-scale, focusing on smaller features and node-level properties. Many authors reference other methods in their analyses, but never provide adequate reasoning for the difference in results they obtain (Przulj *et al.* , 2004; Middendorf *et al.* , 2005; Przulj, 2007; Su *et al.* , 2011). Thus a more in-depth look at each method is necessary to determine why different results are being obtained and if one method provides a better empirical network classification than the others.

In this chapter, we present five network classifiers. Three methods, relative graphlet frequency, RGF, (Przulj *et al.* , 2004), graphlet degree distribution using arithmetic mean, GDD (A), and graphlet degree distribution using geometric mean, GDD (G) (Przulj, 2007) all compare graphs based on small-scale properties. Characteristic curve (Su *et al.* , 2011) , CC, works with large-scale properties. We also propose a new classifier, degree distribution distance, which also references large-scale properties. This latter classifier, abbreviated as DDD, is based on a common network classification idea, however the exact algorithm used is a novel one.

Each of these methods was designed to classify protein-protein interaction networks. Both variations of GDD classified fourteen high-throughput eukaryotic PPI networks (Przulj, 2007). This list of PPI networks includes *Sacchromyces cerevisiae* (baker's yeast), *Drosophila melanogaster* (fruit fly), *Caenorhabditis elegans* (nematode worm), and human. Relative graphlet frequency was used to examine both high and low-confidence PPI networks from both the bakers yeast and the nematode worm. The CC was used to examine four different versions of the fruit fly PPI network made with different confidence levels.

After presenting the five classification methods, we conclude by discussing the limitations of the papers that originally presented the four non-novel algorithms.

5.1 Network Classification Methods

5.1.1 Relative Graphlet Frequency and Graphlet Degree Distribution

Przulj created two network classification methods, RGF and GDD. Both rely on the concept of graphlets, Figure 5.1.Graphlets are defined as 'small 3-5 node subgraphs' (Przulj *et al.* , 2004). There are two three-node, six four-node, and twenty-one five-node graphlets, giving a total of 29 graphlets. The term graphlet is used to avoid potential confusion with motif, which is another special type of subgraph. Motifs must be over represented in the graph under investigation compared to a random graph (Shen-Orr *et al.* , 2002).

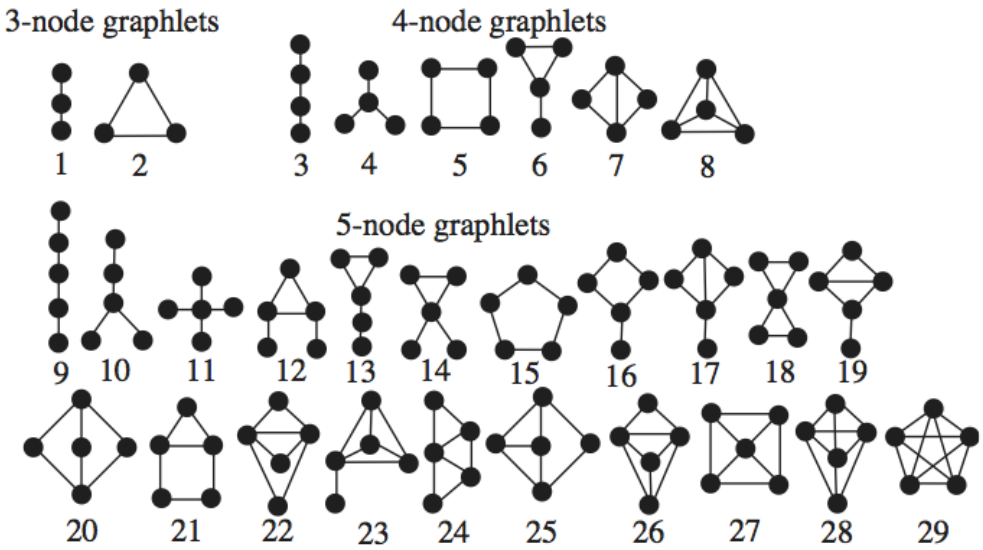


Figure 5.1. **Display of the 29 graphlets (Przulj *et al.* , 2004) - Figure 1.** Graphlets are small 3-5 node subgraphs that can be used to classify networks.

The first method introduced by Przulj is relative graphlet frequency, RGF (Przulj *et al.* , 2004). When measuring relative graphlet frequency, the number of times each graphlet appears is counted. A distance is then determined by:

$$\mathcal{D}_{\mathcal{RGF}}(\mathcal{G}_1, \mathcal{G}_2) = \sum_{i=1}^{29} |F_i(\mathcal{G}_1) - F_i(\mathcal{G}_2)|\,, \tag{5.1}$$

where:

$$F_i(\mathcal{G}) = -log(N_i(\mathcal{G})/T(\mathcal{G})). \tag{5.2}$$

The numerator, $N_i(\mathcal{G})$, is the number of graphlets of type i, $i \in \{1, ..., 29\}$, in graph $\mathcal{G}$, and $T(\mathcal{G}) = \sum_{i=1}^{29} N_i(\mathcal{G})$, the total number of graphlets. The logarithm is used because the frequency of a given graphlet may differ by several orders of magnitude between networks.

The RGF was tested for robustness through random edge additions, deletions, and rewiring. Three percentages of edges were edited: 10%, 20%, and 30%. The method was found to be very robust to edge additions, but only fairly robust to deletions and rewirings at all percentages.

The graphlet degree distribution, GDD, is slightly more complicated than RGF. In this method, each node position of the 29 different graphlets is numbered. This produces 72 unique node positions or automorphism orbits (Figure 5.2). The use of the word position is necessary because in graphlets such as the triangle there is only one unique node position despite the presence of three nodes. In addition to the 72 node positions, another graphlet, with $n = 2$ is added, bringing the total node positions to 73. These node positions are referred to as automorphism orbits and range from 0 to 72.

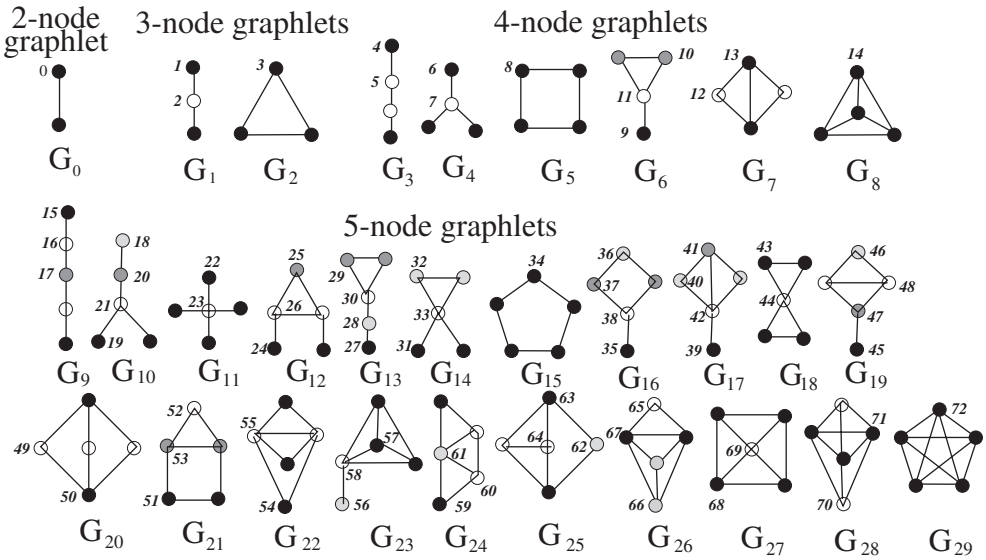


Figure 5.2. **Display of the 73 automorphism orbits (Przulj, 2007)-Figure 1.** Automorphism orbits are unique nodes position within each graphlet. They are differentiated by different color nodes in the image.

For any network $\mathcal{G}$, $d^j_{\mathcal{G}}(k)$ represents the j-th graphlet degree distribution, $j \in \{0, ..., 72\}$. Here j is a particular automorphism orbit, k is the number of times a node acts as the j-th orbit in the network, giving $d^j_{\mathcal{G}}(k)$ as the total number of nodes acting as the j-th orbit k times. It is scaled such that:

$$S^j_{\mathcal{G}}(k) = \frac{d^j_{\mathcal{G}}(k)}{k}. \tag{5.3}$$

This is done to decrease the contribution of larger degrees. The distribution is then normalized with respect to its total area:

$$N^j_{\mathcal{G}}(k) = \frac{S^j_{\mathcal{G}}(k)}{T^j_{\mathcal{G}}}, \tag{5.4}$$

where $T^j_{\mathcal{G}} = \sum_{k=1}^{\infty} S^j_{\mathcal{G}}(k)$. Then $N^j_{\mathcal{G}}(k)$ can be looked at as "the fraction of the total area under the curve, over the entire GDD, devoted to degree k" (Przulj, 2007). The distance between two networks, $\mathcal{G}_1$ and $\mathcal{G}_2$, at a particular automorphism orbit is defined as:

$$D^j(\mathcal{G}_1, \mathcal{G}_2) = \frac{1}{\sqrt{2}} \left(\sum_{k=1}^{\infty} \left[N^j_{\mathcal{G}_1}(k) - N^j_{\mathcal{G}_2}(k) \right]^2 \right)^{1/2}. \tag{5.5}$$

In practice the upper limit of the sum is finite due to the finite size of the network. The distance calculated in Equation 5.5 will always fall between zero and one, with zero implying an identical match at the j-th automorphism orbit between the two networks. In the original 2006 paper, the scaling factor of $\sqrt{2}$ was not included despite the same claim of range (Przulj, 2007). This issue was corrected in an erratum in 2010 (Pržulj, 2010). In order to turn this into an agreement, it is necessary to reverse the values. Thus in order to obtain the GDD agreement at automorphism orbit j we have:

$$A^j(\mathcal{G}_1, \mathcal{G}_2) = 1 - D^j(\mathcal{G}_1, \mathcal{G}_2). \tag{5.6}$$

The overall agreement between two networks then is either the arithmetic (Equation 5.7) or geometric (Equation 5.8) mean of Equation 5.6:

$$A_{arith}(\mathcal{G}_1, \mathcal{G}_2) = \frac{1}{73} \sum_{j=0}^{72} A^j(\mathcal{G}_1, \mathcal{G}_2), \tag{5.7}$$

$$A_{geo}(\mathcal{G}_1, \mathcal{G}_2) = \left(\frac{1}{73} \prod_{j=0}^{72} A^j(\mathcal{G}_1, \mathcal{G}_2) \right)^{1/73}. \tag{5.8}$$

The arithmetic mean is referred to as GDD (A) and the geometric is GDD (G). For more information behind the logic of the design of the agreement measure, see (Przulj, 2007). This method was not evaluated for robustness.

5.1.2 Characteristic Curve

The characteristic curve (CC) by Su *et al.* (Su *et al.* , 2011) is a large-scale classification method. In using the characteristic curve it was necessary to make several assumptions due to ambiguities in the original paper. A characteristic curve for a network is created by choosing a random node to start. It is essential to note at this point that there is not a single characteristic curve for each network, it varies based on the choice of start node. However, the authors state that the process of choosing the start node does not have a significant effect on the outcome. Once a start node has been chosen, all of its neighbors are inserted into a queue at random. The leading node in the queue is then popped (removed) and all of its neighbors are inserted into the queue. Once a node has been popped from the queue the first time it is marked as explored. If the popped node has already been explored, then its neighbors are not added to the queue and the next node is popped. This process continues until all of the nodes have been explored. A pair of coordinates (X, Y) are assigned to each node based on the ratio of its order in the queue to the total number of nodes, X, and the position of the parent copy, Y. The parent copy is the node that brought the node under examination into the queue. The characteristic curve requires a connected network, and thus is only run on the giant component of a network.

In order to compare two networks, a graph distance $\mathcal{D}_{\mathcal{CC}}(\mathcal{G}_1, \mathcal{G}_2)$ is defined as:

$$\mathcal{D}_{\mathcal{CC}}(\mathcal{G}_1, \mathcal{G}_2) = \sum_{X=0}^{\bar{k}} |\mathcal{CC}_1(X) - \mathcal{CC}_2(X)| \frac{1}{2M} \tag{5.9}$$

$$\mathcal{CC}_i(X) = \begin{cases} Y/\bar{k}, & 0 \leq X \leq T_{end} \\ X/\bar{k}, & T_{end} < X \leq \bar{k} \end{cases}, \tag{5.10}$$

where $\mathcal{CC}_i(X)$ represents the characteristic curve of graph i, $i \in \{1, 2\}$, at point X. Notation from the original paper has been slightly edited to conform with the standards introduced at the beginning of this dissertation (Chapter 1). A summary of these standards is available in the Appendix (Table A.1). The general idea of the graph distance is straight forward; it is simply a calculation of the area between two curves. However, Su's notation is ambiguous. The summation goes from $X = 0$ up to the average degree. For our purposes, we assume that the average degree at the top of the summation refers to the maximum average degree of the two graphs being compared such that:

$$\bar{k} = \max\left(\bar{k}_1, \bar{k}_2\right). \tag{5.11}$$

The factor M refers to the maximum number of edges between the two graphs:

$$M = \max\left(m_1, m_2\right), \tag{5.12}$$

where m_i is the total number of edges in network i. For each network, $\mathcal{G}_1$ and $\mathcal{G}_2$, its number of edges dictates the step size of X by increments of $\frac{1}{2m_i}$. Finally, the lower bound of the summation is more clearly written as $X = s_i$ where $s_i \in \mathcal{S}$ and S represents the ordered set of X values of the network with more edges such that $s_1 \leq s_2 \leq \ldots$. The more edges a network has, the smaller each step, and the more steps necessary to traverse the network. The definition for $\mathcal{CC}(X)$ could also be more clear than its representation in Equation (5.10). The definition of $\mathcal{CC}(X)$ is written as a step function which allows us to better match and compare networks with vastly different sizes of giant components (Equation 5.10). In this

equation $\bar{k}$ does not correspond to the overall maximum average degree across to the two network (Equation 5.11, but to the average degree of the given network. The value T_{end} refers to the proportion of nodes in the giant component of the specific network. It is the proportion, as opposed to the overall size, due to how nodes are put into the queue. At the beginning of the distance calculation (Equations 5.9, 5.10) $\mathcal{CC}_i(X) = Y/\bar{k}$. When the whole giant component of a network has been explored, the value of $\mathcal{CC}_i(X)$ is now set to $X/\bar{k}$. This allows networks with drastically different sizes of giant components to be compared. A less ambiguous representation of Equation 5.9 is then:

$$\mathcal{D}_{\mathcal{CC}}(\mathcal{G}_1, \mathcal{G}_2) = \sum_{X=s_i}^{\max(\bar{k}_1, \bar{k}_2)} |\mathcal{CC}_1(X) - \mathcal{CC}_2(X)| \frac{1}{2 \cdot \max(m_1, m_2)} \tag{5.13}$$

$$\mathcal{CC}_i(X) = \begin{cases} Y/\bar{k}_i, & 0 \leq X \leq T_{end,i} \\ X/\bar{k}_i, & T_{end,i} < X \leq \bar{k}_i \end{cases} .$$

As previously mentioned, the characteristic curve only works on fully connected networks. If a network is not connected, the authors deem it acceptable to run the method on the giant component of the network, given that the proportion of nodes in the giant component is acceptably large (e.g., $|\mathcal{V}_{\mathcal{H}}| / |\mathcal{V}_{\mathcal{G}}| > 0.1$). If the giant component is not large enough it will not contain enough of the significant structural features of the network to be considered an accurate representation (Su *et al.* , 2011).

The authors tested the robustness of this classification method using two types of graph perturbations. In the first type of perturbation, a percentage of edges in each graph were randomly replaced. In the second, a percentage of the edges were again rewired, but the degree distribution of the network was held constant. Classification results for these perturbed graphs showed robustness to small and intermediary amounts of noise. Results were optimal for the second type of noise.

5.1.3 Degree Distribution Distance

We propose a novel method based on large-scale network topology. This method, degree distribution distance (DDD), begins by calculating the degree distribution of each network. The degree distribution refers to the number of nodes at each degree in the range from the minimum degree, $\delta(\mathcal{G})$, to maximum degree, $\Delta(\mathcal{G})$, of the whole network, $\mathcal{G}$. Different networks are compared through the use of a distance metric, defined as:

$$\mathcal{D}_{\mathcal{D}}(\mathcal{G}_1, \mathcal{G}_2) = \sum_{k=k_1}^{k_2} |F_k(\mathcal{G}_1) - F_k(\mathcal{G}_2)| \tag{5.14}$$

$$k_1 = min\left(\delta(\mathcal{G}_1), \delta(\mathcal{G}_2)\right)$$

$$k_2 = max\left(\Delta(\mathcal{G}_1), \Delta(\mathcal{G}_2)\right).$$

Thus k_1 is the minimum degree of the two networks being compared and k_2 is the maximum. The value $F_k(\mathcal{G})$ is equal to:

$$F_k(\mathcal{G}) = \begin{cases} -log(N_k(\mathcal{G})/T(\mathcal{G})), & N_k(\mathcal{G}) \neq 0 \\ 0, & N_k(\mathcal{G}) = 0 \end{cases},$$

where $N_k(\mathcal{G})$ is the total number of nodes in graph $\mathcal{G}$ with a degree of k and $T(\mathcal{G})$ is the total number of nodes in $\mathcal{G}$. This definition is comparable to that given by Przujl (Przulj *et al.* , 2004) to describe the distance for her relative graphlet frequency method described in Section 5.1.1. The logarithm of the ratio $N_k(\mathcal{G})/T(\mathcal{G})$ is used here, just as in her paper, since frequencies of degrees can differ by several orders of magnitude between networks (Przulj *et al.* , 2004).

The idea of using degree distribution is not completely unique (Hadley *et al.* , 2012; Wang *et al.* , 2012; Aliakbary *et al.* , 2013), it has not been used in this particular distance metric. None of these formulations tested for classification robustness.

5.2 Limitations of Previous Work

As mentioned at the beginning of this chapter, the methods presented come from several different papers. Since work in this area has already been accomplished, what purpose is there to repeating it? The purpose of our analysis is to address several of the limitations found in previous work. First is the number of different model graphs used. Su *et al.* (Su *et al.* , 2011) only examined three model graphs as potential matches for the empirical network when testing the characteristic curve. These models were DMC, DMR, and LPA (Su *et al.* , 2011). Przulj (Przulj *et al.* , 2004; Przulj, 2007) only looked at four model types for each of her three classifiers. RGF considered RDS, RDS with degree distribution set to match the empirical network, LPA, and GEO (Przulj *et al.* , 2004). She did use three different versions of the GEO model type though, 2-dimension, 3-dimension, and 4-dimension. Both GDD algorithms used the same first three model types as the RGF, but only used GEO-3D (referenced by just GEO in this dissertation).

Considering such a small sampling of the available model graphs does two things. First, it makes it very difficult to compare answers across classifiers. Clearly, if a network is not considered by a paper, then it cannot be chosen as the best fit for the PPI network being classified. Second, with such a small sampling of model graph types, there is a good chance that the best fitting type is not considered. This potentially renders the results of the classification less reliable.

All of the papers also fail to address the issue of large or small-scale categorization. Though the authors do state the category into which their method falls, they do not address limitations due to not examining the other scale. The main ramification of this is the possibility of achieving results that only match on one scale and thus falsely mislabeling a network.

Finally, while Su created 1000 graphs of each model type, Przulj only used 25 in each of her analyses. Given model variability (Chapter 3), this is not necessarily adequate (Burton *et al.* , 2006). In addition, Przulj did not provide evidence of the accuracy of her classifier, thus providing no proof that it has any ability to classify graphs accurately. In the next two chapters, we attempt to rectify the limitations identified in the previous works.

Chapter 6

Random Graph Classification

The first step in the evaluation of any classifier is to determine whether it can perform its job properly. For any classifier, this means examining its ability to accurately classify items into their correct categories. As the first step in examining graph classification, we determine whether any of the five model graphs have the ability to accurately classify random model graphs built with different probabilities, p. This classification analysis considers only accuracy when evaluating each method.

6.1 Methods

The five classification mechanisms were tested on their ability to accurately classify random graphs created with varying probabilities, p. A total of 100 random graphs were created using p in $[0.05, 0.95]$, increasing in increments of 0.05. The graphs were all composed of 250 nodes. This size was chosen because it was large enough for the different probabilities to promote obviously different graphs, while still being small enough to allow for time efficient running and calculation. If the number of nodes in the graph is too small, we increase the likelihood of creating graphs that are not differentiated across the levels of p despite the logical implication that they should be (Kolaczyk & Krivitsky, 2011). This is due to the limited number of possible edges for graphs with few nodes. Of the 100 graphs at each p, 90 were designated as comparison graphs and the remaining 10 were designated as test graphs. The test graphs are the ones that need to be classified.

Every test graph was compared to each of the 90 comparison graphs of the nineteen different p's using the five different classification methods. This led to a total of 1710 comparisons for each test network and a total of 17,100 comparisons per method. A single test graph T_{i,p_t}, where $i = 1, \ldots, 10$ is the number of the test graph and p_t is a specific instance of probability p, is compared to each comparison graph, C_{j,p_c}. The notation, C_{j,p_c}, refers to comparison graph where $j = 1, \ldots, 90$ and p_c in a specific instance of probability p.

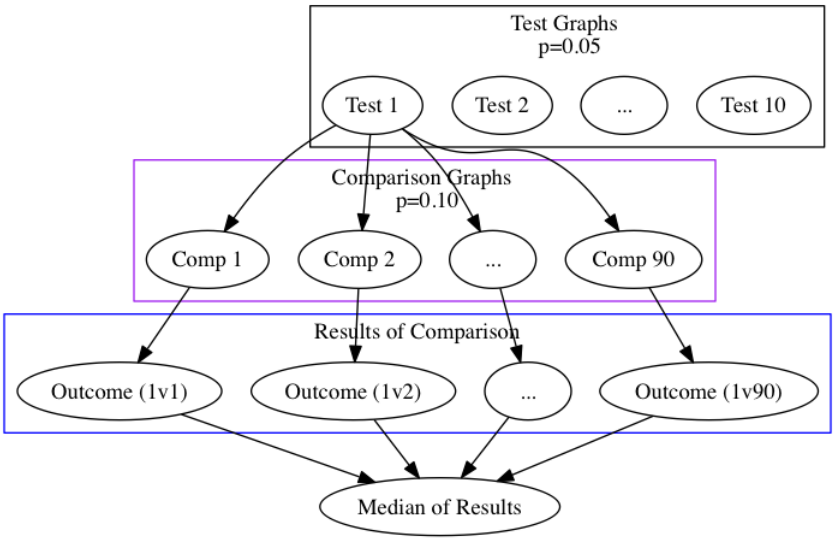


Figure 6.1. **Example of random graph classification procedure: comparison step.** Here we see how the first test graph (Test 1 in black box) created by using $p = 0.05$ is compared to all graphs created by using $p = 0.10$ (purple box). Then the outcomes of each comparison are presented (blue box). The median of these results is then taken. The whole process must be repeated for test graphs 2 - 10.

Each comparison results in an outcome value. In Figure 6.1 these outcomes are labeled such that "Outcome (1v1)" refers to the comparison of test network 1 to comparison network 1. From these 90 outcome values, the median result is calculated. This process is repeated with the same test graph being compared to comparison graphs created with the remaining probabilities. This process results in a list of nineteen median results for test graph 1 (Figure 6.2).

From the list of median results, the best resulting value is chosen based on the method used for comparison. If DDD, CC, or RGF is utilized, the best resulting value is the smallest value since all of these methods calculate a distance. If the method is either GDD, which use an agreement instead of a distance, the best fit is the largest value. The probability of the comparison graphs, p_c, corresponding to the best resulting value is determined the best fit for test network 1 at p_t. Once the best fit for test network T_{1,p_t} has been declared, the process is repeated for the remaining test networks, 2 through 10, created with the same probability, p_t. If p_t is equal to the best fit p_c, then the test network was properly classified. If the two probabilities are not equal, then the test network was incorrectly classified. Once the best fit for all T_{i,p_t}, $i = 1, \ldots, 10$, has been calculated, it is

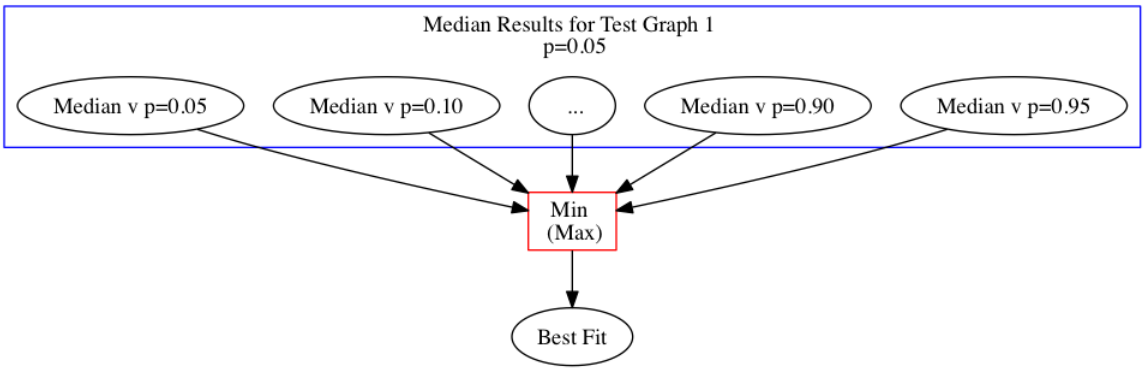


Figure 6.2. **Example of random graph classification procedure: best fit step.** Once the procedure in Figure 6.1 has been repeated for comparison levels of each p, the median results are accumulated (blue box). The minimum value, or maximum in the event that the classifier is either GDD, is determined. The p of the graphs resulting in this minimum (maximum) value is deemed the best fitting random graph for the test graph. If the creation probabilities for both the test graph and the best fitting comparison graph match, then the test graph was accurately classified. The procedure is repeated for each test graph of a single probability in order to determine the average classification accuracy for that random model type.

possible to determine the proportion that were correctly fit. The whole process is repeated for every test network for every probability.

6.2 Results

When degree distribution distance was used as the classification mechanism, it classified every model graph correctly. The relative graphlet frequency did the same. The characteristic curve, however, did not perform nearly as well.

Using the median outcome values, 70 graphs ($\sim 37\%$) were classified incorrectly by the CC. Random networks built with $p \leq 0.60$ had a better record of classification than those with $p > 0.60$ (Table 6.1). It should be noted that there is a large, unexpected drop in classification accuracy for $p = 0.70, 0.75, 0.85$. None of the test graphs in these categories were classified correctly by CC. This is interesting, because graphs with created with a 5-10% difference in probability of edge connection had high classification accuracies. This outcome, therefore, appears erroneous. If we examine Table 6.2, however, we can see that the misclassified graphs are never placed into groups more than 5-10% different than their correct one.

Table 6.1. Classification accuracy of random graphs using the characteristic curve.

Model (p_t)	**Classification Accuracy % Correct**
0.05	100
0.10	100
0.15	100
0.20	100
0.25	80
0.30	60
0.35	90
0.40	60
0.45	60
0.50	70
0.55	70
0.60	60
0.65	30
0.70	0
0.75	0
0.80	70
0.85	0
0.90	90
0.95	40
Average	**62%**

The model column indicates the value of p with which the graphs were designed. The second column provides the classification accuracy of the ten graphs classified for each of the nineteen model categories.

The two versions of the graphlet degree distribution, based on arithmetic and geometric mean respectively, did not accurately classify all of the random graphs. Their

Table 6.2. Full description of classifications of random graphs using the characteristic curve.

		Predicted Class																		
		0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.8	0.85	0.90	0.95
Actual Class	0.05	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0.10	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0.15	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0.20	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0.25	-	-	-	10	**80**	10	-	-	-	-	-	-	-	-	-	-	-	-	-
	0.30	-	-	-	-	-	**60**	40	-	-	-	-	-	-	-	-	-	-	-	-
	0.35	-	-	-	-	-	-	**90**	10	-	-	-	-	-	-	-	-	-	-	-
	0.40	-	-	-	-	-	-	20	**60**	20	-	-	-	-	-	-	-	-	-	-
	0.45	-	-	-	-	-	-	-	30	**60**	10	-	-	-	-	-	-	-	-	-
	0.50	-	-	-	-	-	-	-	-	20	**70**	10	-	-	-	-	-	-	-	-
	0.55	-	-	-	-	-	-	-	-	-	30	**70**	-	-	-	-	-	-	-	-
	0.60	-	-	-	-	-	-	-	-	-	-	20	**60**	20	-	-	-	-	-	-
	0.65	-	-	-	-	-	-	-	-	-	-	-	30	**30**	40	-	-	-	-	-
	0.70	-	-	-	-	-	-	-	-	-	-	-	20	30	**-**	40	10	-	-	-
	0.75	-	-	-	-	-	-	-	-	-	-	-	-	-	50	**-**	50	-	-	-
	0.80	-	-	-	-	-	-	-	-	-	-	-	-	-	10	-	**70**	-	20	-
	0.85	-	-	-	-	-	-	-	-	-	-	-	-	-	-	30	40	**-**	30	-
	0.90	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**90**	10
	0.95	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	60	**40**

The values presented are the percent of random graphs classified into each category by the characteristic curve. Values in bold indicate the percentage of the graphs from each class that were correctly classified. Note that in the majority of misclassifications, graphs are only misclassified by 5%. Only a few graph types, $p \in [0.70, 0.85]$, have graphs misclassified by 10%.

performance was a significant improvement over the CC, however. Both GDD versions misclassified three graphs made at $p = 0.95$. Of the three graphs labeled incorrectly, two were labeled as $p = 0.70$ and the remaining was $p = 0.65$. These results were the same for both GDD (A) and GDD (G), though the actual agreement values differed slightly. This resulted in an accuracy of 98% for both GDD classifiers.

Figure 6.3 shows a visual of the distances and agreements of the random graph classification by four of the five classifiers. The plot show results for only one of the ten test graphs from each of the nineteen types, however differences across the ten tests graphs are minute. Figure 6.3a and Figure 6.3c, the plots for the DDD and the RGF respectively, show clearly that the graphs are classified correctly. It is interesting to note that in both situations, the minimum distances are not directly correlated with the change is p. That is graphs built with large p are not more (or less) likely to have more extreme minimum agreements than those built with small p. Both visuals appear very symmetric in this respect. This is in sharp contrast to Figure 6.3d, the plot for the classification of random graphs using the arithmetic version of the GDD. The geometric version is not shown because it is redundant.

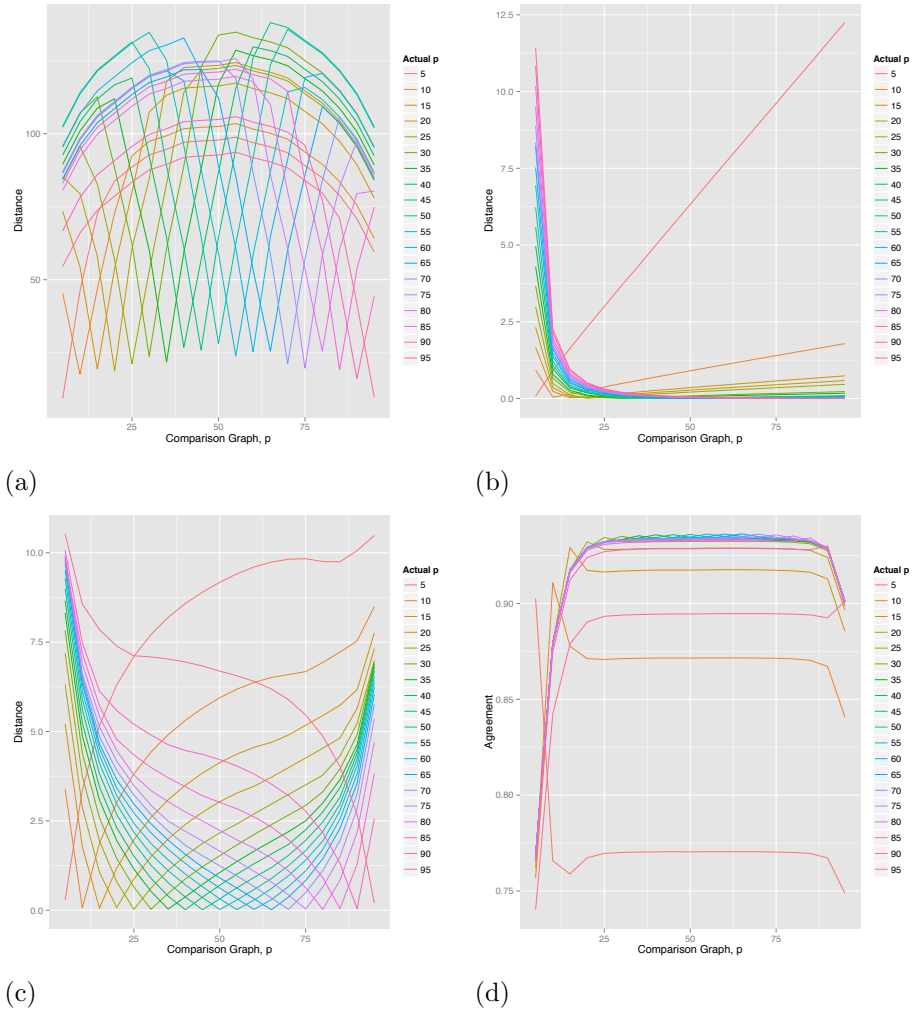


Figure 6.3. **Visualization of random graph classification.** The x-axis is p (multiplied by 100). The y-axis is the distance (or agreement in the case of GDD). The plots show the classification distances for one of the ten test graph from each of the nineteen random graph types (p). Each individual line is for a test graph showing the median classification distance (y-axis) for the test graph compared to the comparison graph (x-axis). **(a):** Result for the degree distribution distance. All of the test graphs are classified correctly. This can be seen because the minimum distance for each test graph, p_t, occurs when $p_t = p_c$. Minimum distances, for the best fitting model types, and maximum distances, for the worst fitting, do not vary much across the different test graphs. Test graphs created with larger probabilities do not have correspondingly maximum larger distances. **(b):** Result for the characteristic curve. With the exception of the test graphs $p = 0.05, 0.10$, all of the test graphs display their largest values when compared to graphs with small p_c. Test graphs created with larger p_t do have correspondingly larger maximum distances. **(c):** Results for relative graphlet frequency. Similar to **(a)**, 100% of the test graphs were classified correctly. Graphs created with more extreme probabilities display larger fluctuations in distance than those made with more centered probabilities. Minimum distances do not change significantly across the different model graph categories. **(d):** Result for graphlet degree distribution using arithmetic mean. Results are not shown for the geometric mean because they are redundant. Agreement values are more extreme for graphs created with low probabilities. The highest agreement value is also substantially lower for these test graphs. The differences between agreements of graphs made with probabilities above about 0.25 do not vary significantly. There is simply a small peak at the best fit. This lack of difference is further show in Table 6.3.

In this latter figure, graphs created with different probabilities of edge creation have different agreements, and the differences form a pattern. The correct choice is still very obvious in each case, however the density of the graph appears to have an effect on the maximum agreement achieved (Table 6.3). All of the graphs were classified accurately in this situation. The lowest agreements (0.9026, 0.9012) occur at $p = 0.05$ and $p = 0.95$ respectively. The highest agreement achieved is 0.9363 at $p = 0.65$. Another interesting feature to note is that the same agreement values, up to four significant figures, are seen multiple times. This is true for 0.9362, which is achieved $p = 0.55, 0.60$, and 0.70, and for 0.9358, which is achieved at $p = 0.40$ and 0.75.

Further, we note that for GDD there are very small differences between the first and second highest agreement levels (Figure 6.3 and Table 6.3) . Many of these values do not differentiate themselves until the third significant digit. The biggest differences are seen on the high and low end of the edge creation probabilities.

Table 6.3. Graphlet degree distribution using arithmetic mean random graph classification comparison of first and second place results.

	Agreement		
p_t	**1st**	**2nd**	$p_{c,2}$
0.05	0.9026	0.7705	0.45, 0.55, 0.60, 0.65, 0.70
0.10	0.9108	0.8777	0.15
0.15	0.9291	0.9176	0.60
0.20	0.9322	0.9289	0.55, 0.60, 0.65
0.25	0.9344	0.9325	0.60
0.30	0.9350	0.9335	0.35
0.35	0.9353	0.9340	0.40
0.40	0.9358	0.9343	0.45
0.45	0.9359	0.9345	0.50
0.50	0.9360	0.9347	0.55
0.55	0.9362	0.9350	0.60
0.60	0.9362	0.9350	0.65
0.65	0.9363	0.9349	0.60
0.70	0.9362	0.9349	0.65
0.75	0.9358	0.9346	0.70
0.80	0.9352	0.9338	0.75
0.85	0.9342	0.9325	0.55, 0.60, 0.65, 0.80
0.90	0.9299	0.9287	0.60, 0.70
0.95	0.9012	0.8947	0.60, 0.65, 0.70

Table shows a comparison of the best and second best agreement values obtained using GDD (A). The first column indicate the p with which the model was made. The second column gives the agreement value of the best matching model type. The third column gives the second best agreement value. The $p_{c,1}$ of the best matching model type is not given because these were classified accurately. The $p_{c,2}$ of the second best matching model type are given. In some cases multiple p_c achieved this second best values. Values are obtained from only one of the ten test graphs from each of the nineteen types. It is the same set of test graphs as seen displayed in Figure 6.3d. It is clear that for the non-extreme values of p, the differences between first and second place agreements are essentially negligible.

Finally, the plot shown in Figure 6.3b displays the distance results for the CC. Most of the lines follow a pattern similar to the plot of $y = \frac{1}{x}$, $x > 0$. The highest distance calculated occurs at $p = 0.05$, then there is a sharp drop off before the distances slowly begin to increase again. In this situation, the differences between the smallest and second smallest distance are extremely small just like the GDD. Once again, they do not differ until the third significant figure in many cases.

6.3 Discussion

In this chapter, we began the steps to evaluate the capabilities of the five classification mechanisms. We built 100 random model graphs for each edge creation probability in the range $[0.05, 0.95]$, increasing by increments of 0.05. Of the 100 graphs of each type, 10 were designated test graphs that needed to be classified. The remaining 90 were designated comparison graphs. These were used as known examples for comparison with the test graphs.

Results showed that both RGF and DDD performed with perfect accuracy when classifying random model graphs. We speculate that this is because certain graphlets, and certain node degrees, have a minimum amount of connectivity required to exist. For instance, it is impossible for graphlet 29 to exist if there are not ten edges. Therefore, these specific features may ease the classification burden.

Both variations of the GDD performed very well. Only three random graphs were misclassified. Since these were each created with $p = 0.95$, we speculate that at the highest level of probability, the distinction between different graphs is drastically reduced.

The characteristic curve is the only method that performed poorly in classifying random graphs. The higher classification success rate of the CC on graphs built with probabilities of edge connection in the range $[0.05, 0.60]$ may imply that the CC works more accurately with graphs of lower density. This is backed up by the perfect classification of test graphs built with $0.05 \leq p \leq 0.20$ and the far less than perfect classification of graphs built with $p \geq 0.65$. The improved performance for graphs of lower densities may be an acceptable flaw of the CC because the real world networks for which this method was designed to classify typically have low densities. The difference in classification accuracies

can also imply that when the creation probability increases to a certain level, such as $p = 0.65$, the differences between random graphs are not as clear and straightforward as they are for lower probabilities.

Overall, none of the classification methods performed so poorly in their classification of random graphs that their results are unexplainable nor their failures unforgiveable. This is true even for the CC because the real world networks we are looking to classify have low densities. Unfortunately, it is likely that the PPI networks that we are interested in classifying are not random graphs at all, but rather fall into a more complex category of model graphs. Therefore, this step is not fully adequate in deciphering the abilities of the classifiers. We must test the classifiers abilities on non-random model graphs. This step is performed in the next chapter (Chapter 7).

Chapter 7

Model Graph Classification

The papers from which the relative graphlet frequency (Przulj *et al.* , 2004) and both graphlet degree distribution measures (Przulj, 2007) were introduced neglected to provide any validation that their methods can accurately classify known model graphs into their correct categories. This is an essential missing piece because without it, it is impossible to determine the reliability of the results. Therefore, in this chapter we test the ability of each classifier to correctly assess a set of known model graphs. We evaluate the results by accuracy, like in Chapter 6, but add in several other statistics: F-measures, positive predictive value (PPV), negative predictive value (NPV), sensitivity, and specificity. We conclude the chapter by offering the next steps in this analysis.

7.1 Methods

Out of the 1000 graphs simulated for each of the nine models, 900 were designated comparison graphs and the remaining 100 were test graphs. The process of comparison was the same as for the random graphs (Figures 6.1 and 6.2). The DDD is the only classification algorithm that was able to use all of the DMC and DMR graphs. The other methods used a sampling of the graphs with less than 50,000 edges. For RGF and GDD, this was necessary because GraphCrunch 2 does not work on networks above this size (Kuchaiev *et al.* , 2011). For CC, while it is technically possible to run this method on large networks, it poses severe time restrictions. There were 782 DMC networks with less than 50,000 edges and 789 DMR networks. In both of these situations 100 networks were still designated as test graphs and the remaining as comparison graphs.

Each method was evaluated in several ways. First, overall accuracy was calculated to provide a general idea of how many graphs were classified correctly. Then a confusion matrix was created. A confusion matrix is table layout that allows one to visual the performance of a classifier (Stehman, 1997). Each confusion matrix can be collapsed into nine separate 2x2 matrices such as Figure 7.1, one for each of the model graph types. From these, the sensitivity (ρ), specificity, positive predictive value (π), negative predictive value, and F-

measure (Equation 7.5) can be calculated for each model type. The sensitivity, or recall, of the classifier identifies the algorithms ability to classify a model type correctly (Loong, 2003). It is calculated for the i^{th} model type, $i \in \{1, ..., 9\}$, by:

$$\rho_i = \frac{TP}{TP + FN}. \tag{7.1}$$

In other words, this is the proportion of graphs that are classified as Type A that actually are Type A.

The specificity of a classifier identifies its ability to not misclassify other graphs as that particular model type, or the proportion of graphs not classified as Type A that really are type A:

$$\text{SPC}_i = \frac{TN}{TN + FP}. \tag{7.2}$$

The positive predictive value (PPV), or the precision, is:

$$\pi_i = \frac{TP}{TP + FP}. \tag{7.3}$$

This tells the proportion of graphs classified as Type A that actually are Type A. A negative predictive value (NPV) is the proportion of graphs that are correctly classified as Not Type A that are actually Not Type A:

$$\text{NPV}_i = \frac{TN}{TN + FN}. \tag{7.4}$$

Finally, the F-measure (i.e. F_1 score or F-score), was calculated for each model type. The F-measure is a measure of a binary classification test's accuracy. This was chosen over other potentially more informative statistics because it can be modified to assess the accuracy of multi-class classifiers, as opposed to simply binary. Unfortunately, this measure does not take the true negative into account, which may lead to instances of miscommunication.

		Predicted Class: A	Predicted Class: Not A
True Class	A	True positive (TP)	False negative (FN)
	Not A	False positive (FP)	True negative (TN)

Figure 7.1. **Example binary confusion matrix.** A confusion matrix is a visual way of displaying the results of a classification.

The F-measure for a single model type i is calculated by:

$$F_i = \frac{2\pi_i\rho_i}{\pi_i + \rho_i}. \tag{7.5}$$

This is the harmonic mean of precision and recall. To calculate the effectiveness of the entire algorithm across all of the model types, the F-measure can be modified to become the micro-average F-measure and macro-averaged F-measure (Özgür *et al.* , 2005). In the micro-averaged F-measure (F-micro) the precision and recall are calculated globally by summing over the true positive, false positive, and false negative values (Equations 7.6, 7.7).

$$\begin{aligned} \rho &= \frac{TP}{TP + FN} \\ &= \frac{\sum_{i=1}^{9} TP_i}{\sum_{i=1}^{9}(TP_i + FN_i)} \end{aligned} \tag{7.6}$$

$$\begin{aligned} \pi &= \frac{TP}{TP + FP} \\ &= \frac{\sum_{i=1}^{9} TP_i}{\sum_{i=1}^{9}(TP_i + FP_i)} \end{aligned} \tag{7.7}$$

Then, F-micro is:

$$F\text{-micro} = \frac{2\pi\rho}{\pi + \rho}. \tag{7.8}$$

Equation 7.8 is considered an average over each model type because they are all given equal weight. It can be dominated by certain high performing types (Özgür *et al.* , 2005).

In the macro-averaged F-measure (F-macro), the F-measures computed for each model type in Equation 7.5 are averaged together. Thus, F-macro is:

$$F\text{-macro} = \frac{\sum_{i=1}^{9} F_i}{9}. \tag{7.9}$$

All of the F-measure statistics fall between zero and one, with larger values indicating better performance.

For all of the statistics for each classifier, average values of sensitivity, specificity, PPV, and NPV were calculated. These values were averaged from the specific values derived for each model type separately. We also examined the global values, obtained by adding all of the true positives, true negative, false positives, and false negatives for each measure. The statistics were then calculated on these global values resulting in global summary statistics.

7.2 Results

7.2.1 Filtering Out Large Graphs

As previously mentioned, only the DDD was able to use all of the DMC and DMR graphs created due to classification algorithm size constraints. The remaining classifiers were unable to work effectively on graphs larger than 50k edges (Kuchaiev *et al.* , 2011; Su *et al.* , 2011). In order to determine whether this might have a significant effect on the classification outcome, a comparison of these two model graph types based on a sampling of graph measures can be seen in Table 7.1. It should be noted at this point that all of the classification algorithms have ways of comparing graphs of markedly different sizes. These ways ensure that model graphs are not penalized for not being the same size as the network being classified.

Though the median number of edges decreases drastically when only the smaller networks are considered, the number of nodes only decreases by about 100 ($\approx 7 - 8\%$) for both DMC and DMR. The values for most of the other features do not vary greatly when the larger networks were eliminated. The biggest change is for the assortativity values ($r(\mathcal{G})$) of

the DMR graphs. When all 1000 models were considered, there are signs of disassortativity. However when only graphs with less than 50k edges were considered, signs of assortativity are shown instead. Since the yeast PPI network shows signs of dissasortativity, this change may contribute to the DMR being incorrectly not chosen as the best fit. From the results of the comparison, we can conclude that while the final results of any classification may be altered due to the different sampling of graphs, the change in comparison models is unavoidable. In addition, it should have minimal effect on the outcome because so few graph measures displayed any change.

Table 7.1. Comparison of all DMC, DMR graphs to those with less than 50k edges.

	DMC		DMR	
	$< 50k$	All	$< 50k$	All
Full Network				
m	2321	4703.5	2429	4746
n	1270	1324	1132	1237
$\mathcal{D}$	0.0021	0.0038	0.0024	0.0078
$C(\mathcal{G})$	0.144	0.1607	0.0085	0.0114
$\lvert\mathcal{V}_{\mathcal{H}}\rvert \,/\, \lvert\mathcal{V}_{\mathcal{G}}\rvert$	0.5333	0.2624	0.972	0.9936
$\mathcal{S}(\mathcal{G})$	0.7555	0.7754	0.6158	0.6469
$r(\mathcal{G})$	0.0661	0.0312	0.0266	-0.0343
Giant Component				
Diameter	12	9	12	9
Radius	6	5	6	5
$\bar{\ell}$	4.578	3.6209	4.4923	3.7238
$\log n$	2.8031	2.9901	3.0374	3.0883
$\log\log n$	0.4476	0.4757	0.4825	0.4897
$\mathcal{S}(\mathcal{G})$	0.7515	0.7726	0.6137	0.6411
$r(\mathcal{G})$	-0.0461	-0.0685	-0.0345	-0.0689

Table reports results of several network measures for the set of all DMC (DMR) graphs as well as for only those with less than 50k edges. This gives an idea of the loss of information that occurs due to size constraints found in CC, RGF, and GDD.

7.2.2 Degree Distribution Distance

In Chapter 6, we noted the degree distribution distance's impressive ability to correctly classify random graphs. Unfortunately, it did not prove quite so impressive when attempting to classify the model graphs (Table 7.2).

DDD was only able to correctly classify one model type, RDG, 100% of the time. Three other models types were classified correctly more than two-thirds of the time: SMW,

Table 7.2. Classification accuracy of degree distribution distance.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
True Class	AGV	9	-	-	9	-	82	-	-	-
	DMC	-	1	-	7	-	12	4	46	30
	DMR	-	2	2	8	-	12	8	30	38
	GEO	-	-	-	73	-	-	27	-	-
	LPA	-	-	-	-	-	100	-	-	-
	RDG	-	-	-	-	-	100	-	-	-
	RDS	-	-	-	3	-	-	97	-	-
	SMW	-	-	-	-	-	-	1	99	-
	STI	-	-	-	-	-	75	-	-	25

The values presented are the percent of model graphs classified into each category by DDD.

RDS, and GEO. The remaining five graph types were classified correctly less than a quarter of the time. In fact, one graph, LPA, was never correctly classified. All of the LPA graphs were classified as RDG.

Table 7.2 can also be presented as a visual confusion matrix. An example of a perfect confusion matrix can be seen in Figure 7.2. The plot shows the ideal red squares on the diagonal indicating that 100% of all graphs were classified correctly. The results from Table 7.2 can be seen in Figure 7.3. Perfect classification is obviously not the case for the results of DDD classification. Most of the model graphs that were classified incorrectly, were grouped between one or two incorrect choices. For instance, the incorrect AGV graphs were classified as either GEO and RDG, with most falling in the latter category. The DMC and DMR graphs, however, were spread out in their incorrect classification. The overall classification accuracy of the DDD was only 45% (Table 7.13).

Table 7.3 displays a statistical analysis of the DDD's performance. In an ideal world, we would desire all of the values in the table to be near one. In actuality, there are often trade-offs. A high sensitivity is sacrificed for a high level of specificity and vice-versa, which is visualized in Figure 7.4. Most of the values that begin with a low sensitivity, seen on the third axis, increase significantly to a high specificity as seen on the fourth axis.

The trade-off in specificity and sensitivity is very evident in the classification of LPA. It has zero sensitivity because it classified none of the model LPA graphs correctly, however it has very high specificity because it also didn't incorrectly classify any graphs as LPA, Table 7.2. In addition, RDS and SMW scored high in every category with the exception of

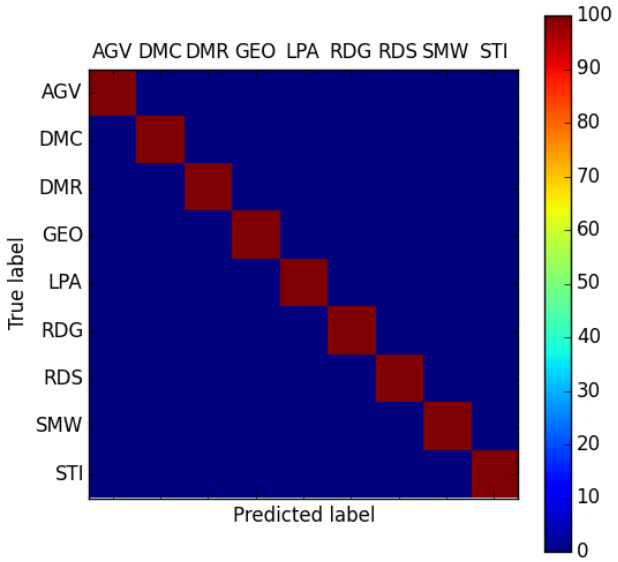


Figure 7.2. **Example of a confusion matrix displaying perfect classification.** The red squares on the diagonal show that 100% of the graphs were classified into the correct category.

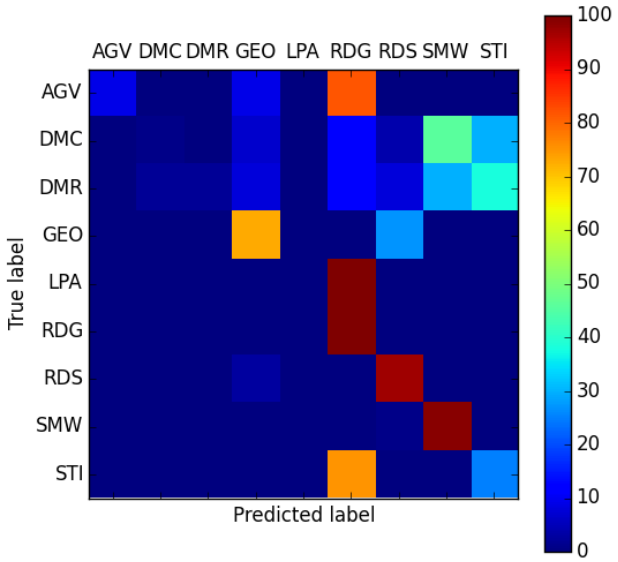


Figure 7.3. **Degree distribution distance classification results confusion matrix.** Image shows how all of the the test graphs of each type were classified. A total of 100 graphs of each model type were used. Red squares indicate higher classification accuracy, blue squares indicate lower.

Table 7.3. Degree distribution distance statistical analysis of performance.

Model	PPV	NPV	Sensitivity	Specificity	F-measure
AGV	1.0	0.8979	0.09	1.0	0.1651
DMC	0.333	0.8896	0.01	0.9975	0.0194
DMR	1.0	0.8909	0.02	1.0	0.0392
GEO	0.73	0.9663	0.73	0.9663	0.73
LPA	0.0	0.8889	0.0	1.0	0.0
RDG	0.2625	1.0	1.0	0.6488	0.4158
RDS	0.708	0.9961	0.97	0.95	0.8186
SMW	0.5657	0.9986	0.99	0.9050	0.72
STI	0.2688	0.9071	0.25	0.9150	0.2591
Average	0.5409	0.9373	0.4511	0.9314	0.3519
Global	0.4511	0.9999	0.4511	0.9314	0.4511

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the DDD. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

PPV. This indicates that they are often the choice for incorrectly classified models. They are not nearly as popular as RDG, which has 57% of the incorrectly classified graphs (281 out of 494). That is also why RDG has extremely low PPV and specificity despite being the only model graph that was classified 100% correctly. Therefore, the DDD is inadequate at correctly telling which networks are LPA, as corroborated by the PPV and F-measure, however it is perfect at telling which graphs are not LPA, as seen by NPV.

We can conclude that DDD is better at telling us what type of model a graph is not, than what it is. This conclusion is based on the significantly higher NPV than PPV. While this feature is not ideal, it does have the potential to be useful. The F-macro, the average of all the individual model graph F-measures, is 0.3519 which indicates that this is not a reliable method. The F-micro is marginally better at 0.4511.

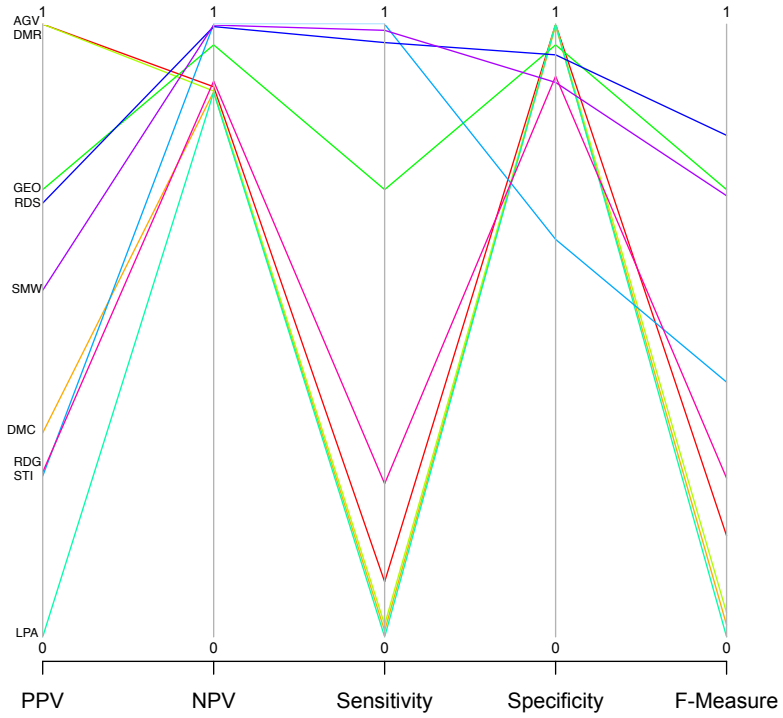


Figure 7.4. **Parallel coordinate representation of the degree distribution distance performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 100 models of each of the nine model graph types.

7.2.3 Characteristic Curve

The characteristic curve is unique in its requirement of a connected network. In order to get an idea of how using only the giant component affects features of a graph, the full network and giant component of the PPI network were compared to each other using the three other classifiers considered in this chapter (Table 7.4).

Table 7.4. Comparison of *S. cerevisiae* PPI network full network v giant component.

	DDD	RGF	GDD (A)	GDD (G)
Distance or Agreement	1.3468	0.0004	0.9917	0.9915

Characteristic curve requires a connected component in order to run, thus it is typically run on the giant component. This table shows the full network of the *S. cerevisiae* PPI network compared to the giant component by the four other classifiers. Lower values are better for DDD and RGF. Values closer to one are better for both forms of GDD.

For DDD and RGF, smaller numbers indicate smaller distances and thus better matches. For both forms of GDD, the closer a number is to one, the better the agreement. From Table 7.4 we conclude that based on the other methods, it is valid for the CC to run its comparisons using only the giant component as it does not appear to lose many of its features. This is based on the distance and agreement values resulting from comparing the full network to its giant component using the DDD, RGF, and both GDD. This also confirms the assertions made by Su, *et al.*, that a large enough giant component retains significant features of the full network, where large enough is defined as greater than 10% of the nodes in the full network are present in the giant component (Su *et al.* , 2011). This latter criterion is easily met by the model graphs.

It was seen in the previous chapter that the characteristic curve performed poorly at differentiating between random graphs with different densities and it did not perform much better when classifying the nine model graphs (Table 7.5). Three model types were classified incorrectly 100 percent of the time: AGV, DMC, and DMR. The remaining networks were all classified correctly at least 50 percent of the time, with four graphs classified correctly at or above 98 percent. Overall, 58 percent of the model graphs were classified correctly.

Table 7.5. Classification accuracy of characteristic curve.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
True Class	AGV	-	-	-	9	30	-	22	39	-
	DMC	-	-	-	3	32	4	-	-	61
	DMR	-	-	-	7	30	7	1	-	55
	GEO	-	-	-	100	-	-	-	-	-
	LPA	-	-	-	-	98	-	2	-	-
	RDG	-	-	-	-	-	98	-	-	2
	RDS	-	-	-		-	-	100	-	-
	SMW	-	-	-	10	-	-	14	76	-
	STI	-	-	-	-	-	50	-	-	50

The values presented are the percent of model graphs classified into each category by CC.

Incorrectly classified graphs were typically spread over two to three different models (Figure 7.9). The majority of these misclassifications are into LPA, RDG, SMW, or STI. Just as no AGV, DMC, or DMR graphs were correctly classified, no models were incorrectly classified into any of these categories either.

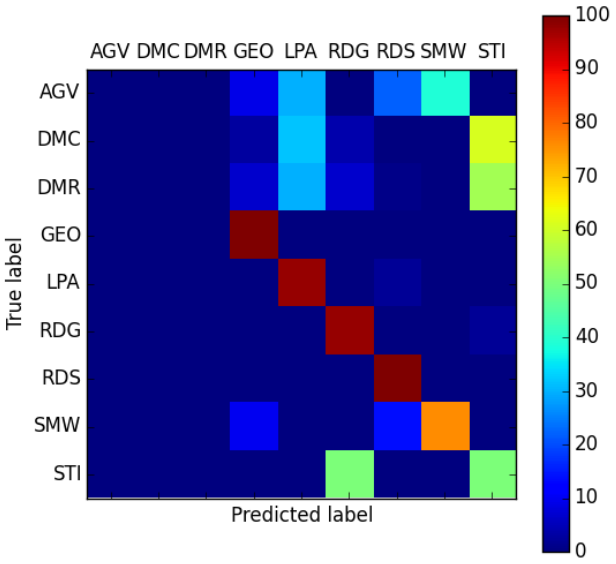


Figure 7.5. **Characteristic curve classification results confusion matrix.** Image shows how all of the the test graphs of each type were classified. A total of 100 graphs of each model type were used. Red squares indicate higher classification accuracy, blue squares indicate lower.

Table 7.6 and Figure 7.6 display the statistical analysis of the CC. There are three patterns visible in Figure 7.6. The first, in yellow, is the plot of AGV, DMC, and DMR.

These models have a sensitivity, PPV, and F-measure of zero and then very high specificity and NPV values, 0.9 and 1.0 respectively. The second pattern, consisting of STI and SMW, follows a very similar pattern to the first but without the extreme peaks and valleys. They both have lower values for PPV, sensitivity, and F-measure with higher NPV and specificity. The third pattern begins towards the top of the figure. Here, GEO, RDS, RDG, and LPA all have a relatively high PPV that increases to a higher NPV. The values then stay roughly constant for the sensitivity. They all show a slight drop for specificity, though the decrease is of varying sizes, and continue the drop for the F-measure. Overall, these four model types perform well across all statistics and their overall performance is indicated in the F-measures which range from 0.68 to 0.87.

Overall, the average specificity and NPV are both close to one, however their relevance is brought into question by the low average sensitivity of 0.58 and the even lower PPV, Table 7.6, Figure 7.6. Global values are higher for PPV (0.58 v 0.3984) and NPV (0.9999 v 0.9514). The F-macro is 0.4692 and F-micro is 0.58 both confirming than this model still leaves much to be desired in terms of correct classification.

Table 7.6. Characteristic curve statistical analysis of performance.

Model	**PPV**	**NPV**	**Sensitivity**	**Specificity**	**F-measure**
AGV	0.0	0.8889	0.0	1.0	0.0
DMC	0.0	0.8889	0.0	1.0	0.0
DMR	0.0	0.8889	0.0	1.0	0.0
GEO	0.7752	1.0	1.0	0.9638	0.8734
LPA	0.5158	0.9972	0.98	0.8850	0.6759
RDG	0.6164	0.9973	0.98	0.9238	0.7568
RDS	0.7194	1.0	1.0	0.9513	0.8368
SMW	0.6609	0.9694	0.76	0.9513	0.707
STI	0.2976	0.9317	0.5	0.8525	0.3731
Average	0.3984	0.9514	0.58	0.9475	0.4692
Global	0.58	0.9999	0.58	0.9475	0.58

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the CC. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

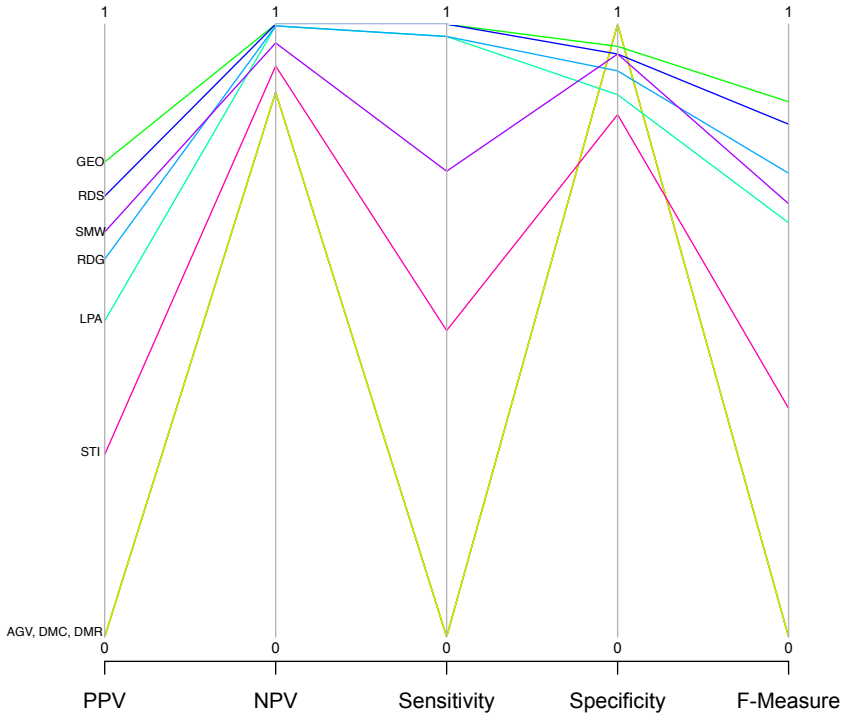


Figure 7.6. **Parallel coordinate representation of the characteristic curve performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 100 models of each of the nine model graph types.

7.2.4 Relative Graphlet Frequency

The RGF classified five model graphs correctly 100% of the time, however it failed on the remaining four (Table 7.7). Only thirteen AGV and two SMW model graphs were classified correctly. The misclassified AGV graphs were mostly classified as LPA, though 19 were RDG (Figure 7.7). SMW graphs were spread out in their misclassification, hitting six of the nine model categories. They were never classified as DMC, DMR, or STI. None of the DMC or DMR graphs were correctly classified. DMC was classified as everything except DMC or DMR with a slight majority of networks (33%) classified as RDG. DMR was less spread out in its misclassification with 48 classified as RDS and 35 as SMW. The overall classification accuracy for RGF is 57%.

Table 7.7. Classification accuracy of relative graphlet frequency.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
True Class	AGV	13	-	-	-	68	19	-	-	-
	DMC	4	-	-	11	2	33	29	1	20
	DMR	-	-	-	1	-	16	48	35	-
	GEO	-	-	-	100	-	-	-	-	-
	LPA	-	-	-	-	100	-	-	-	-
	RDG	-	-	-	-	-	100	-	-	-
	RDS	-	-	-	-	-	-	100	-	-
	SMW	22	-	-	14	17	4	41	2	-
	STI	-	-	-	-	-		-	-	100

The values presented are the percent of model graphs classified into each category by RGF.

Based on the statistical analysis of the relative graphlet frequency, this measure also does a poor job classifying graphs (Table 7.8). It has approximately equal F-macro and F-micro, both at about 0.45. High specificity and NPV are once again negated by poor PPV and sensitivity. All of the model types can be separated into two groups based on Figure 7.8. The first group contains STI, GEO, RDG, LPA, and RDS. These five models have higher sensitivity than specificity with both values greater than 0.9. Their PPV are all above 0.5. The F-measures for these models are between 0.63 and 0.89 indicating that we can conclude with reasonable certainty that graphs classified into these groups are accurate. The remaining models, AGV, DMC, DMR, and SMW are in the second group. These graphs all have extremely low sensitivities and PPV. The classification failures represented

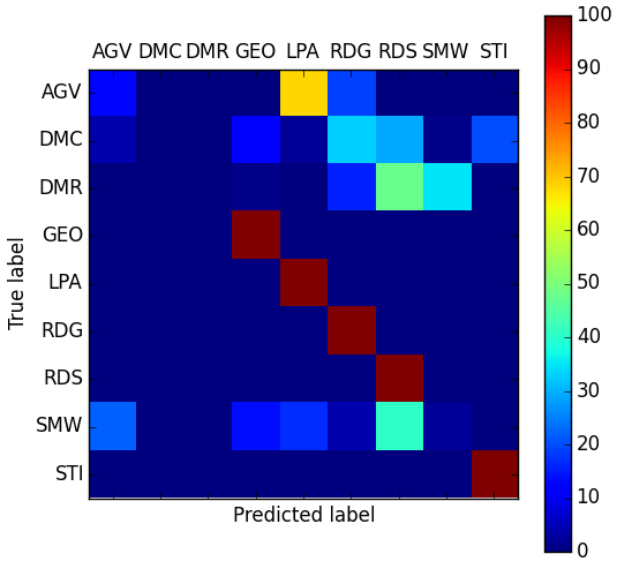


Figure 7.7. **Relative graphlet frequency classification results confusion matrix.** Image shows how all of the the test graphs of each type were classified. A total of 100 graphs of each model type were used. Red squares indicate higher classification accuracy, blue squares indicate lower.

in those values are reflected in F-measures that are barely above zero.

Table 7.8. Relative graphlet frequency statistical analysis of performance.

Model	**PPV**	**NPV**	**Sensitivity**	**Specificity**	**F-measure**
AGV	0.3333	0.899	0.13	0.9675	0.1871
DMC	0.0	0.8889	0.0	1.0	0.0
DMR	0.0	0.8889	0.0	1.0	0.0
GEO	0.7937	1.0	1.0	0.9675	0.8850
LPA	0.5348	1.0	1.0	0.8913	0.6969
RDG	0.5814	1.0	1.0	0.91	0.7353
RDS	0.4587	1.0	1.0	0.8525	0.6289
SMW	0.0526	0.8863	0.02	0.955	0.0290
STI	0.833	1.0	1.0	0.9750	0.9091
Average	0.3986	0.9514	0.5722	0.9465	0.4524
Global	0.5722	0.9999	0.5722	0.9465	0.5722

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the RGF. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

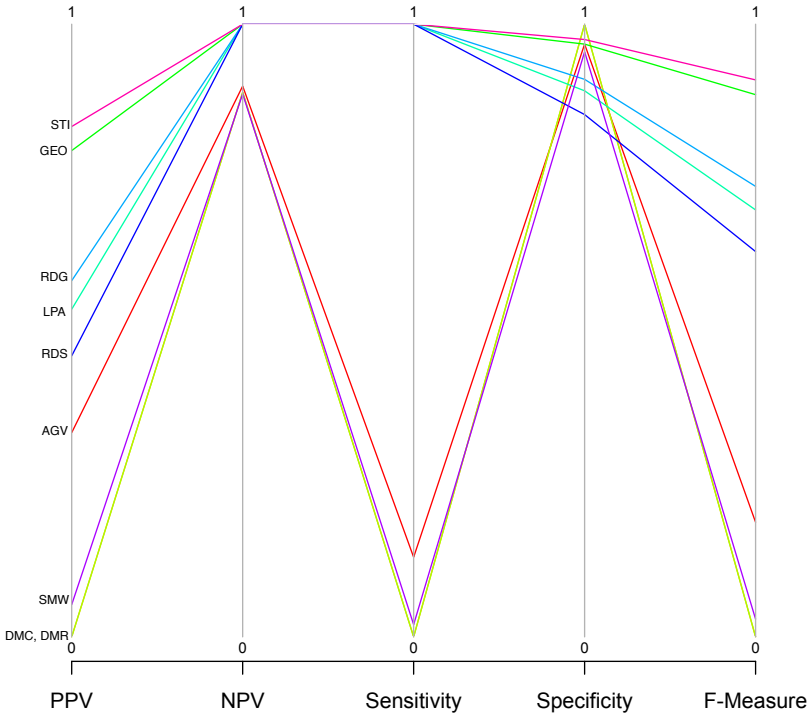


Figure 7.8. **Parallel coordinate representation of the relative graphlet frequency performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 100 models of each of the nine model graph types.

7.2.5 Graphlet Degree Distribution

Two versions of the graphlet degree distribution were considered. One calculated the average distribution using the arithmetic mean and the other used the geometric mean. The former method was more accurate, 68% v 60% (Tables 7.9, 7.10). The results across the two versions are very similar with the main difference in accuracy being found in the classification of LPA. In GDD (G) only half of the LPA graphs were correctly classified while they all were in GDD (A) (Tables 7.9, 7.10). When models were incorrectly classified in this method they were only relegated to two other model types. There is not a large spread across the incorrect results (Figures 7.9a, 7.9b).

Table 7.9. Classification accuracy of graphlet degree distribution using arithmetic mean.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
True Class	AGV	40	-	-	40	20	-	-	-	-
	DMC	-	-	-	90	-	-	10	-	-
	DMR	-	-	10	-	-	10	70	-	10
	GEO	-	-	-	100	-	-	-	-	-
	LPA	-	-	-	-	100	-	-	-	-
	RDG	-	-	-	-	-	100	-	-	-
	RDS	-	-	-	-	-	-	100	-	-
	SMW	-	-	-	30	-	-	10	60	-
	STI	-	-	-	-	-		-	-	100

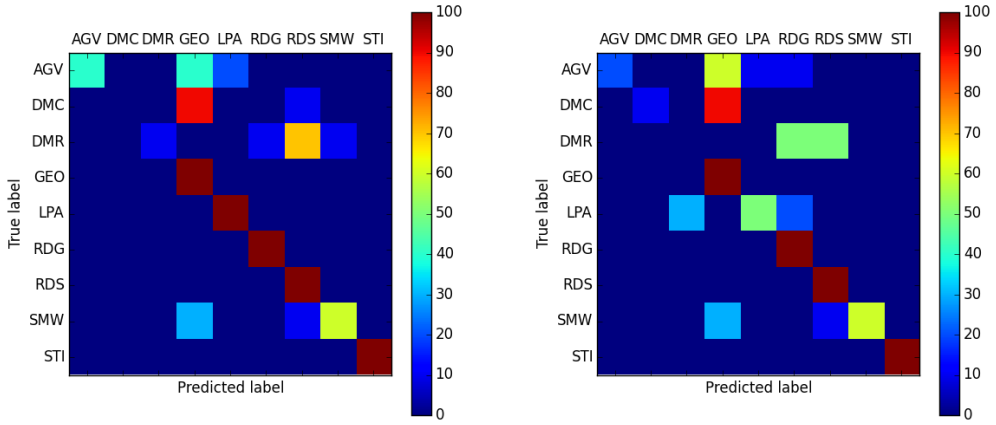
The values presented are the percent of model graphs classified into each category by GDD (A).

Table 7.10. Classification accuracy of graphlet degree distribution using geometric mean.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
True Class	AGV	20	-	-	60	10	10	-	-	-
	DMC	-	10	-	90	-	-	-	-	-
	DMR	-	-	-	-	-	50	50	-	-
	GEO	-	-	-	100	-	-	-	-	-
	LPA	-	-	30	-	50	20	-	-	-
	RDG	-	-	-	-	-	100	-	-	-
	RDS	-	-	-	-	-	-	100	-	-
	SMW	-	-	-	30	-	-	10	60	-
	STI	-	-	-	-	-		-	-	100

The values presented are the percent of model graphs classified into each category by GDD (G).

As previously mentioned, the classification results for the two GDDs are extremely similar. This is largely seen in the performance analyses of the methods, but there are a few differences worth noting (Tables 7.11, 7.12). The largest, and expected difference, is



(a) (b)

Figure 7.9. **Graphlet degree distribution classification results confusion matrix.** Image shows how all of the the test graphs of each type were classified. A total of 100 graphs of each model type were used. Red squares indicate higher classification accuracy, blue squares indicate lower. **(a):** Graphlet degree distribution using arithmetic mean. **(b):** Graphlet degree distribution using geometric mean.

that of the AGV values. All of the values under the arithmetic mean are larger since it classified more AGV correctly and neither method incorrectly classified any graphs as AGV. A slightly unexpected difference is seen in the RDG values. Under both methods, 100% of the RDG graphs were correctly classified. This is directly shown through the perfect NPV and sensitivity. The values that are very different are the PPV. The arithmetic version produced a PPV of 0.9091 for RDG while the geometric mean produced 0.5556. This is because under the arithmetic mean only 10 graphs were incorrectly classified as RDG. Under the geometric mean 80 graphs were. Thus the former method more accurately classifies graphs as RDG. In the latter method, nearly half of its classifications were incorrect. A similar, though marginally less drastic, result is seen in the F-measure (0.9524 v 0.7143). The rest of the images (Figures 7.10, 7.11) are nearly identical. The F-macro for GDD (A) is 0.618 and the F-micro is 0.6778. For GDD (G) those values are 0.5444 and 0.6. Global values for GDD (A) are also higher than those for GDD (G) with the exception of NPV where the two are equal.

Table 7.11. Graphlet degree distribution using arithmetic mean statistical analysis of performance.

Model	PPV	NPV	Sensitivity	Specificity	F-measure
AGV	1.0	0.9302	0.4	1.0	0.5714
DMC	0.0	0.8889	0.0	1.0	0.0
DMR	1.0	0.8989	0.1	1.0	0.1818
GEO	0.3846	1.0	1.0	0.8	0.5556
LPA	0.8333	1.0	1.0	0.975	0.9091
RDG	0.9091	1.0	1.0	0.9875	0.9524
RDS	0.5263	1.0	1.0	0.8875	0.6897
SMW	1.0	0.9524	0.6	1.0	0.75
STI	0.9091	1.0	1.0	0.9875	0.9524
Average	0.7292	0.9634	0.6778	0.9597	0.618
Global	0.6778	0.9999	0.6778	0.9597	0.6778

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the GDD (A). Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

Table 7.12. Graphlet degree distribution using geometric mean statistical analysis of performance.

Model	PPV	NPV	Sensitivity	Specificity	F-measure
AGV	1.0	0.9091	0.2	1.0	0.3333
DMC	1.0	0.8989	0.1	1.0	0.1818
DMR	0.0	0.8851	0.0	0.9625	0.0
GEO	0.3571	1.0	1.0	0.775	0.5263
LPA	0.8333	0.9405	0.5	0.9875	0.625
RDG	0.5556	1.0	1.0	0.9	0.7143
RDS	0.6250	1.0	1.0	0.925	0.7692
SMW	1.0	0.9524	0.6	1.0	0.75
STI	1.0	1.0	1.0	1.0	1.0
Average	0.7079	0.9540	0.6	0.95	0.5444
Global	0.6	0.9999	0.6	0.95	0.6

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the GDD (G). Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

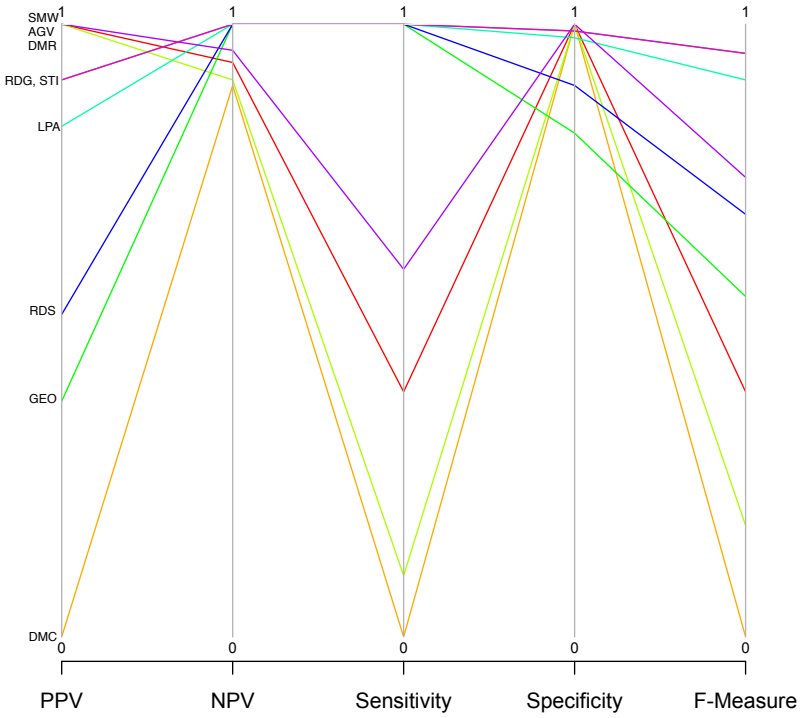


Figure 7.10. **Parallel coordinate representation of the graphlet degree distribution using arithmetic mean performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 100 models of each of the nine model graph types.

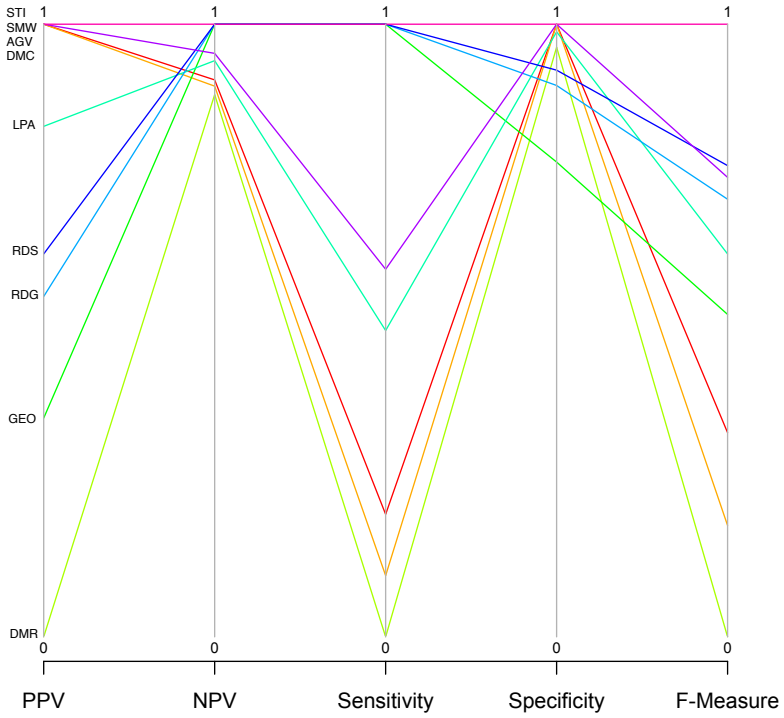


Figure 7.11. **Parallel coordinate representation of the graphlet degree distribution using geometric mean performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 100 models of each of the nine model graph types.

7.2.6 Comparison of Classification Accuracy Broken Down by Model Type and Method

An overview of all of the classification mechanisms' accuracy reveals disappointing results (Table 7.13). GDD (A) performed the best with 68% of test graphs classified accurately while DDD was worst with 45% of the 900 test graphs classified correctly.

Table 7.13. Comparison of the classification accuracy breakdown by model type and method.

	Classification Accuracy (% Correct)					
Model	DDD	CC	RGF	GDD (A)	GDD (G)	**Average**
AGV	9	0	13	40	20	**16.4**
DMC	1	0	0	0	10	**2.2**
DMR	2	0	0	10	0	**2.4**
GEO	73	100	100	100	100	**94.6**
LPA	0	98	100	100	50	**69.6**
RDG	100	98	100	100	100	**99.6**
RDS	97	100	100	100	100	**99.4**
SMW	99	76	2	60	60	**59.4**
STI	25	50	100	100	100	**75**
Average	**45**	**58**	**57**	**68**	**60**	

The values in the table indicate the percentage of the given model graph that was accurately classified by the classification method.

Looking at a breakdown of classification results by model type, it is interesting to note that some model types were clearly very easy, or very difficult, to classify because nearly all of them were classified correctly, or incorrectly, by all five of the methods. The easy to classify model types include GEO (94.6%), RDG (99.6%), and RDS (99.4%) while the difficult model types are AGV (16.4%), DMC (2.2%), and DMR (2.4%). The remaining three model types are a bit of a mixture. LPA graphs were classified correctly 69.6% of the time, but most of the misclassification came from DDD. This method classified no LPA graphs correctly. For SMW graphs, these were more easily classified by the methods looking at large-scale features, DDD and CC, than by those looking at small scale. If we consider only the large-scale methods, the classification accuracy increases to 87.5% versus 40.67%. The opposite is true for the STI graphs. These were much more easily classified by the methods considering small-scale features than large-scale (100% v 37.5%).

There are several additional features that we can use to compare the classification methods: F-macro, F-micro as well as both average and global sensitivity, specificity, PPV, and NPV (Figure 7.12). The lines in Figure 7.12 do not cross indicating that a classifier

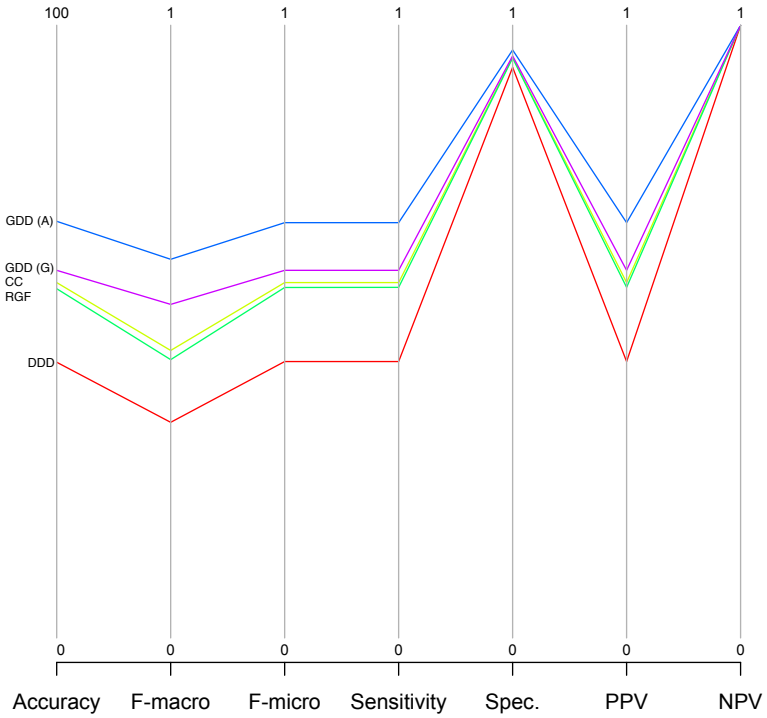


Figure 7.12. **Parallel coordinate comparison of classification method performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and both F-measures. Values presented for sensitivity, specificity, PPV and NPV are the global results of the classification results of the 100 models of each of the nine model graph types.

that is better in terms of one feature is better in terms of all of them. The values tend to group together for specificity and NPV.

There are a few other interesting things to note. The CC has the most model graph types that were not correctly classified even a single time: AGV, DMC, and DMR, whereas GDD (A), GDD (G), and DDD each only misclassified one graph type incorrectly every single time, DMC, DMR, or LPA respectively. The RGF incorrectly classified DMC and DMR every time. The majority of the misclassified graphs were incorrectly classified as RDG (21%) with GEO coming in second (18%).

7.2.7 Patterns in Statistical Performance

The results of the classification method validation show an interesting trend in the statistical analysis of the methods (Figures 7.4 - 7.11). Model graph types can be classified into one of four groups based on their results, Figure 7.13. In Group 1, the models begin with a relatively high PPV compared to the others displayed. Their values increase for the NPV and then stay approximately constant for the sensitivity. A slight decrease of varying sizes is seen for the specificity followed by a larger increase for the F-measure. The second group, Group 2, has very low PPV, sensitivity, and F-measure. These low values are punctuated by high peaks at the NPV and specificity. Groups 3 and 4 are very similar to Group 2. Group 3 typically has the high PPV seen in Group 1 with the rest of the values following the same trend in Group 4. The models seen in Group 4 follow the same overall pattern as Group 2 except transposed to the middle of the plot and with smaller variation between peaks.

The models that fall into Group 1 are those that are nearly all classified correctly. Unfortunately, these models also tend to be popular choices for incorrect classifications, thus the decrease in their PPV and specificity. The more popular a choice a model is as an incorrect answer, the more the model is penalized in these values.

The reason for the second pattern is that very few, sometimes even zero, of the model graphs in this group were classified correctly, but at least a few other graphs were incorrectly placed into their categories. If no models were incorrectly placed into a category with low accuracy, then the model falls into Group 3. The incorrect classification of all the

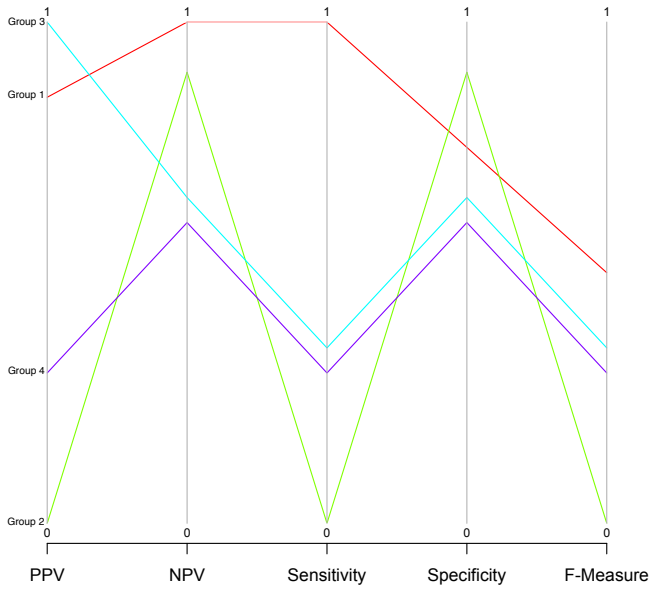


Figure 7.13. **Example parallel coordinate representation of model performance statistics indicating group classifications.** It was found that each parallel coordinate plot showed four patterns. These patterns can be used to easily visually group model graph types. Each group has a set of features unique to it. **Group 1:** high accuracy, popular incorrect choice. **Group 2:** low accuracy, unpopular, but not negligent, incorrect choice. **Group 3:** low accuracy, negligibly chosen incorrectly. **Group 4:** moderate to high accuracy, moderate to low popular incorrect choice.

graphs in these categories provides the zero for sensitivity. The lack of models incorrectly placed into these categories leads to a very high, or in Group 3 perfect, but trivial, specificity and high NPV.

The models that fall into Group 4 tend to have moderate to high correct classification accuracy. They also have a moderate to low number of incorrect classification in their category. Overall, the groups can be summarized as follows.

- Group 1 - high accuracy, popular incorrect choice
- Group 2 - low accuracy, unpopular, but not negligent, incorrect choice

- Group 3 - low accuracy, negligibly chosen incorrectly
- Group 4 - moderate to high accuracy, moderate to low popular incorrect choice

Table 7.14. Model graph groupings by patterns in performance statistics.

	Group 1	Group 2	Group 3	Group 4
DDD	**RDG**, **RDS**, SMW	LPA, DMC, STI	AGV, DMR	GEO
CC	LPA, **RDG**, **RDS**, SMW	-	AGV, DMC, DMR	GEO, STI
RGF	GEO, LPA, **RDG**, **RDS**, STI	DMC, DMR, SMW	-	AGV
GDD (A)	GEO, LPA, **RDG**, **RDS**, STI	DMC	AGV, DMR, SMW	-
GDD (G)	GEO, **RDG**, **RDS**, STI	DMR	AGV, DMC, SMW	LPA

There are four unique patterns that appear repeatedly in the parallel coordinate plots showing performance statistics. This table shows how each model graph type is grouped by each classifier. The model types in bold appear in only one group across all five classifiers. Groups correspond to those seen in Figure 7.13.

In Table 7.14 we can see that models tend to stick to similar categories even across the different classification algorithms. Thus despite the differences seen, it is clearly apparent from the implications of the interpretations of the groups, and from Table 7.13, that some models are easier to classify than others. GEO, RDG, and RDS are by far the easiest. Their average classification across the schemes is 94.6, 99.6, and 99.4% respectively. These models appear mostly in Group 1. RDG and RDS appear only in Group 1. DMC and DMR have very low overall accuracy, 2.2 and 2.4% respectively. Thus, they typically appear in either Group 2 or Group 3, the groups characterized by low accuracy. Grouping depends on whether any other models were incorrectly classified as that model. It is possible to be fairly confident of any the classification of any real-world network in Groups 1 or 4. Naturally, it does depend on the exact values of all the parameters. Classification into Groups 2 or 3 cannot be counted on for accuracy with any certainty.

7.2.8 Treatment of DMC and DMR Model Graphs

Another interesting trend seen across the five classifiers is the treatment of the DMC and DMR model graphs. No method classified more than 10 of either correctly and only one method, DDD, classified a few of both. CC and RGF both classified all of these graphs incorrectly. In addition, many of these incorrect classifications were not relegated to only one or two incorrect categories. The average number of classes to which DMC was classified is 4.2 per method and DMR was 4.4. The next highest were AGV and STI, both with 3.4.

The overall average number of classes to which a model was classified is 2.4. The median is 1.6.

Table 7.15. Comparison of the classification accuracy breakdown by model type and method where DMC, DMR are not included.

	Classification Accuracy (% Correct)					
Model	DDD	CC	RGF	GDD (A)	GDD (G)	**Average**
AGV	9	0	13	60	36	**23.6**
GEO	73	100	100	100	94	**93.4**
LPA	0	98	100	100	71	**73.8**
RDG	100	98	100	100	100	**99.6**
RDS	97	100	100	100	100	**99.4**
SMW	99	76	2	47	49	**54.6**
STI	25	50	100	100	97	**74.4**
Average	**58**	**75**	**74**	**87**	**78**	

The values in the table indicate the percentage of the given model graph that was accurately classified by the classification method. Five classification methods are shown. DMC/DMR were not classified, nor were they used as potential options for other models to be classified as. This is to show that these model types were the main failure of all the measures.

If we consider Table 7.15 which contains the accuracies of the classification methods when DMC and DMR are not considered, we see that these are all substantially larger values than in Table 7.13. In fact, the GDD (A) is almost at 90% accuracy, a point where any results of real-world network classification could be trusted. While there is still room for improvement, it seems that the methods are almost good at classifying the other model types.

It is concerning that none of the classification methods placed these models with any semblance of accuracy since these models have repeatedly been found to be the best fit for PPI networks (Su *et al.* , 2011; Middendorf *et al.* , 2005; Ispolatov *et al.* , 2005; Pastor-Satorras *et al.* , 2003), however a subtle point must be raised. Is it the fault of the classifiers that they cannot predict these models well, or is there something going on with the models? We conclude that it is a little bit of both. In Chapter 10, we discuss improvements to one of the classifiers that improves its classification accuracy of DMC and DMR, then we provide a novel classifier in Chapters 11, 12, and 13 that does an even better job. Finally, in Chapter 14, we present some theories as to what is going on with these model types and propose several ways to deal with it.

7.3 Discussion

In this chapter we tested the abilities of the five classifiers under investigation by looking at their ability to accurately classify known model graphs. Of the 1000 simulated model graphs from each of the nine types, 100 were designated as test graphs. These were the graphs that the classifiers had to place into their correct groups. The remaining 900 graphs were the comparison graphs. The test graphs were compared against these during the classification procedures. We then analyzed the results using accuracy, F-measures, PPV, NPV, sensitivity, and specificity.

The discrepancy between the results seen in the success of the DDD's classification abilities for the random graph analysis and the failure seen in the model graph validation can be attributed to the fact that this method only utilizes one graph characteristic and this characteristic is associated with graph density. Random graphs created with different probabilities all have noticeably different densities, leading to noticeable differences in degree distribution. Thus, based on the poor model graph classification, we can infer that the degree distributions between the types of models are not that different despite known topological differences. It was, however, very unexpected that 57% of the 494 incorrectly classified graphs were classified as RDG.

Su (Su *et al.* , 2011) was the only author of the considered classifiers that showed results demonstrating the accuracy of their classifier. In Su's paper, they tested classification accuracy on only four model graphs: LPA, DMC, DMR, and RDS. Only the former three graphs were compared to the PPI network. Classification was found to be near perfect. There were a few DMC graphs misclassified as DMR and vice versa, but all the LPA and RDS graphs were correctly classified. When the same process for method validation was performed here, the results were not duplicated (Table 7.16). None of the DMC or DMR graphs were correctly classified. All DMC graphs were classified as LPA. Most DMR networks were also LPA, though a few were RDS. All of the RDS networks were classified correctly, as were the majority of LPA. Two of the latter model graphs were mistakenly classified as RDS.

There are several reasons that the discrepancies seen in Table 7.16 may have occurred. This is the table that compares the model graph validation results as performed

Table 7.16. Attempts at reproducing classification accuracy of the original presentation of the characteristic curve analysis with four model graphs.

	Classification Accuracy (%)[1]			
Original	DMC	DMR	LPA	RDS
DMC	(99.0)	(1.0)	100.0	-
DMR	(3.4)	(95.7)	97.0	3.0 (0.9)
LPA	-	-	98.0 (100.0)	2.0
RDS	-	-	-	100.0 (100.0)

Values in parentheses are from the original CC classification seen in Su *et al.* (Su *et al.* , 2011). Values outside of the parentheses are from tests performed in this dissertation. The values presented are the percent of model graphs classified into each category by the given classifier.

by Su to those performed for this analysis. First, there were fewer model validation comparisons performed here. Su *et al.* used a thousand test networks compared to another thousand classification networks. This is significantly more than the number used here, though the 100 test networks and 900 comparisons should be statistically large enough to achieve convergence to an accurate distance (Burton *et al.* , 2006).

Another potential reason for the disagreement is the size of the PPI network. This paper utilized the *S. cerevisiae* PPI network with 1361 nodes and 3222 edges. Su used three different versions of the *Drosophila melanogaster* PPI network. These versions were based on a different confidence threshold, where each interaction is given a confidence score based upon the likelihood that an interaction will occur *in vivo*. If a confidence score is greater than the confidence threshold then the interaction remains in the model. This method is used to limit the number of false positives. Confidence thresholds of 0.65, 0.50 and 0.0 were used by Su. The resulting *Drosophila melanogaster* PPI networks had 3,279/4,508/6,823 nodes and 2,728/ 4,569/19,630 edges respectively. Since all model graphs are based directly off of the PPI network in question, this results in different sized model graphs.

A final potential reason for these discrepancies is the ratios of nodes to edges in the networks. In the *Drosophila melanogaster* PPI networks have node to edge ratios of 1.20/0.97/0.35 respectively. The *S. cerevisiae* PPI network has a node to edge ratio of 0.42. This is much closer to the ratio seen when the confidence threshold for the *Drosophila melanogaster* PPI network is set to zero. If the results presented in the table are based upon graphs created to match networks created with confidence thresholds of 0.65 or 0.50, then these discrepancies could explain the difference in outcomes. In addition, later results

in Su's paper imply that the networks based on the first two confidence thresholds have different forms than when they are built on a confidence threshold of zero.

The huge drop off in accuracy for the RGF and the GDD between the random graph analysis and model graph validation is likely due to reasons similar to those for the DDD. Just like the degree distribution, the formation of certain graphlets requires a certain number of edges, or a certain density. With the random graphs, the densities are all very different across probabilities, thus it can be assumed that the number and type of graphlets counted is distinctive. The densities across many of the model graphs are very similar, indicating that there may not be distinctive graphlet fingerprints for each model type.

Final results of the comparisons of the five classifier indicate that while GDD (A) has the best classification algorithm consider, it is only the best from a bad set of choices. Average specificities and NPV are high, while average sensitivities and PPV are low. This is to be expected because when each classifier's results are broken down by model type into the nine separate binary classifications, it is more likely for a model to be correctly not placed into the category than to be correctly placed because there are more graphs that are not of the model type. That is why the global values for the NPV show higher values. Unfortunately these values are still not high enough to indicate good classification results.

Within the global values of the statistics we also see that the F-micro is always equal to the average and global specificities, as well as the global PPV. This is because globally, there are an equal number of false positive and false negatives.

We discussed that DMC and DMR graphs are treated differently than the other types of model graphs. They are significantly more likely to be classified incorrectly; just over 2% of each type were classified correctly across the five classifiers. They are also more likely to be classified into numerous different categories while other models typically are misclassified into only one or two incorrect model types. These observations, along with the wide variation seen in graph measures, lead us to conclude that these growth mechanisms do not produce consistent model graphs. The main reason for this is due to the lack of constraint on the number of edges. As previously mentioned, DMC and DMR models create and remove edges based upon uniformly random values p and q. While similar mechanisms are used in other models such as RDG and RDS, the latter methods take the

desired number of edges as an input. Without a required number, network sizes range from 21 edges to over 880k. This large variation naturally creates variations in graph topology which make it difficult to classify the model graphs accurately,

We are particularly interested in the correct classification of DMC and DMR because these methods were created to model PPIs and thus contain biologically accurate features that are not present in other models. This large variation raises the question of whether we can really say that these networks are all the same type.

7.3.1 Strengths and Limitations

Degree Distribution Distance

The four network classification methods all have their strengths and limitations. The degree distribution distance is the fastest of the four methods. This was also the only method that did not require any changes to be made to the model graphs. It works well on unconnected networks and, while logically there is an upper limit in the size of the graph that it can handle, this limit was never reached in our analyses. Thus, this classification method was the only one that was able to use all of the DMC and DMR networks, including the largest ones with 880,237 or 537,036 edges respectively. The number of nodes for any graph was never larger than 1361. Besides being fast and more functional on larger networks, this method also correctly classified all of the random graphs. Unfortunately, it did not do well differentiating between different types of model graphs (Table 7.2). In fact, it was only able to classify networks correctly 45 percent of the time when biological networks were considered (Table 7.13). The graphs most often classified incorrectly were those that were often picked as the best fit for PPI networks: DMC, DMR, and STI. Therefore any answer reported using this method could be considered highly suspect. Finally, another limitation for the degree distribution distance is that the distance is not normalized, thus distances calculated on different sets of networks are difficult to compare.

Characteristic Curve

The characteristic curve had far more limitations than strengths. It was extremely slow in performing calculations and had an upper limit in size. The authors (Su *et al.* , 2011)

did not specifically define an upper limit as the creators of GraphCrunch 2 did, however comparisons of networks with $> 50,000$ edges stalled the program. In addition, distance calculations for the CC result in far more mathematical operations than any of the other methods because the step size for the summation is so small. While it is possible that the authors thought this would improve the accuracy, results seem to indicate that this is not the case. The computational complexities of this method result in an extremely slow runtime. The CC performed poorly in its attempts to correctly classify random graphs (Table 6.1) classifying 62 percent correctly. It did perform better at classifying model graphs, which is a problem of greater importance and relevance than random network classification. It was able to correctly classify only 58 percent of the test graphs (Table 7.13). It also correctly classified no DMC or DMR graphs. Thus, once again a classification method has poor classification accuracy on the model graphs that are often chosen as the best representation of PPI networks.

An additional limitation is that this method is not normalized. Thus, it is hard to determine whether a resulting distance is significant or not. As with degree distribution distance, the lack of a normalizing factor can make interpretations difficult as well as prohibit comparisons of distances from different sets of graphs.

A last limitation unique to the CC is that the comparison of networks differs depending on the node picked to start building the curve. While Su determined that this was not a significant factor, as the number of times the *S. cerevisiae* PPI network giant component is compared to itself increases, the distance never converges to zero (Table 7.17). In fact, with the increase in repetitions, there is also an increase in the range of distances reported. In Table 7.17, results are shown for the giant component of the *S. cerevisiae* PPI network compared to itself 1,000, 10,000, and 100,000 times. The start node for the creation of the characteristic curve is chosen randomly for each network. Ideally, the results would all be zero since the two networks are exactly the same, however the start node of the characteristic curve does seem to play a part. Unfortunately, due to the lack of normalization, it is impossible to determine whether these distances reported are large or small.

Table 7.17. Comparison of the *S. cerevisiae* PPI network giant component to itself using the characteristic curve.

Repetitions	Distance Median (Min, Max)
1,000	152.51 (0.0, 460.42)
10,000	155.32 (0.0, 518.53)
100,000	154.95 (0.0, 550.55)

Table shows median values, as well as minimum and maximum, for the *S. cerevisiae* PPI network giant component compared to itself using the characteristic curve.

Relative Graphlet Frequency

The relative graphlet frequency method has an upper size limit of greater than $50,000$ edges (Kuchaiev *et al.* , 2011). It ran faster than CC, but was not as efficient as DDD. The latter results could be expected because RGF examines smaller details of the networks. Calculations of the number of each of the 29 graphlets in a given network is nontrivial. It is far faster to determine the degree distribution, and, since their distance calculations are essentially the same, DDD is faster. RGF classified 57% of the test graphs correctly (Table 7.13). This method accurately classified all of the STI models, though still incorrectly classified all of the DMC and DMR models. Further limitations of this method are discussed in Chapter 9.

Graphlet Degree Distribution

The graphlet degree distribution has the same upper limit to the number of edges it can handle as RGF. It is also the slowest of all the methods, as well as the most calculation intensive. Its results can be reported using the arithmetic mean (Equation 5.7) or the geometric mean (Equation 5.8). When using the arithmetic mean and not considering biological networks 68 percent of networks were classified correctly (Table 7.13). Using the geometric mean, 60 percent of the models were correctly classified. Thus the arithmetic mean appears to have the highest level of classification accuracy out of all of the classifiers. The GDD using the arithmetic mean was also the only method that classified over 50 percent of the AGV model graphs correctly.

7.3.2 Next Steps

As we have discussed, the results of the classifier validation were not satisfactory. The majority of the issues stem from the methods' attempts to correctly classify the DMC and DMR graphs, though many had significant trouble with AGV and STI graphs as well. Such difficulties indicate a need to revamp the classifiers. In the next several chapters we explore this option. In Chapter 9, we explore how a small mathematical error affects the results of the RGF. In Chapter 10, we propose a faster and more accurate version of the GDD and in Chapters 11 to 13, we propose a novel graph classification algorithm.

Chapter 8

Saccharomyces cerevisiae PPI Network Classification

All of the classifiers discussed in this dissertation that were previously seen in papers were used to classify some organism's PPI network. As we mentioned, one of the main reasons for this work is that these papers did not use a significant sampling of model graph types. Therefore, in this chapter, we examine how the original results of the PPI network classification compare to the results found here. Since it is assumed that PPI networks from all organisms should be classified into the same category, results should be comparable even if the same organism PPI network is not used in both trials (Przulj, 2007). This will allow us to determine whether the authors of the classifiers considered in the previous chapters really did their audiences a disservice by choosing only a limited number of model graph types to be considered.

Since different model types were classified with different levels of accuracy, we begin with an explanation on how we will interpret the classification results. We then present the answers and compare them to the answers found in the original papers. We conclude with a comparison of model graph rankings across all of the classification methods.

8.1 Methods

We classify the *S. cerevisiae* PPI network by comparing it to each of the 1000 model graphs of the nine different types using the five classifiers. This results in a unique list for each classifier ranking all of the model graphs. Results are interpreted using Bayes theorem:

$$\Pr(A \mid B) = \frac{\Pr(B \mid A)\Pr(A)}{\Pr(B)}. \tag{8.1}$$

This theorem describes the probability of an event based on conditions related to the event. In the situation at hand, Equation 8.1 can be interpreted in two ways. First, we can determine that probability that the empirical network is classified as model A given that it is actually model A:

$$\Pr\left(\text{classified as model A} \mid \text{model A}\right) = \frac{\Pr(\text{model A} \mid \text{classified as model A}) \cdot \Pr(\text{classified as model A})}{\Pr(\text{model A})}. \tag{8.2}$$

We can also determine the probability that the empirical network is classified as model A given that it is not actually model A:

$$\Pr\left(\text{classified as model A} \mid \text{NOT model A}\right) = \frac{\Pr(\text{NOT model A} \mid \text{classified as Model A}) \cdot \Pr(\text{classified as model A})}{\Pr(\text{NOT model A})}. \tag{8.3}$$

Probabilities required for the Bayesian analysis are determined from the classification results in Chapter 7.

After calculating the probabilities that the classification of the *S. cerevisiae* PPI network is accurate and determining whether or not the results are reliable, we compare the similarities of the lists. For this we use Kendall's W, also known as Kendall's coefficient of concordance. Kendall's W assesses agreement, but also takes into consideration the number of ranks by which classifiers disagree. For this statistic, we assume that model graph i is given rank $r_{i,j}$ by classifier j where $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$ (Kendall & Smith, 1939). Since we have nine model graphs and five classifiers, $n = 9$ and $m = 5$. Then:

$$R_i = \sum_{j=1}^{5} r_{i,j} \tag{8.4}$$

and the mean of R_i is:

$$\bar{R} = \frac{1}{n} \sum_{i=1}^{9} R_i. \tag{8.5}$$

We then take the sum of squared deviations of the R_i:

$$S = \sum_{i=1}^{n} (R_i - \bar{R})^2, \tag{8.6}$$

so that Kendall's W is defined as follows:

$$W = \frac{12S}{m^2(n^3 - n)} \tag{8.7}$$

$$= \frac{12S}{5^2(9^3 - 9)} \tag{8.8}$$

$$= \frac{12S}{18000}. \tag{8.9}$$

Kendall's W falls between $[0, 1]$, with one representing perfect agreement and zero indicating no trend. If a score of zero is calculated that rankings are essentially random (Li & Schucany, 1975).

8.2 Results

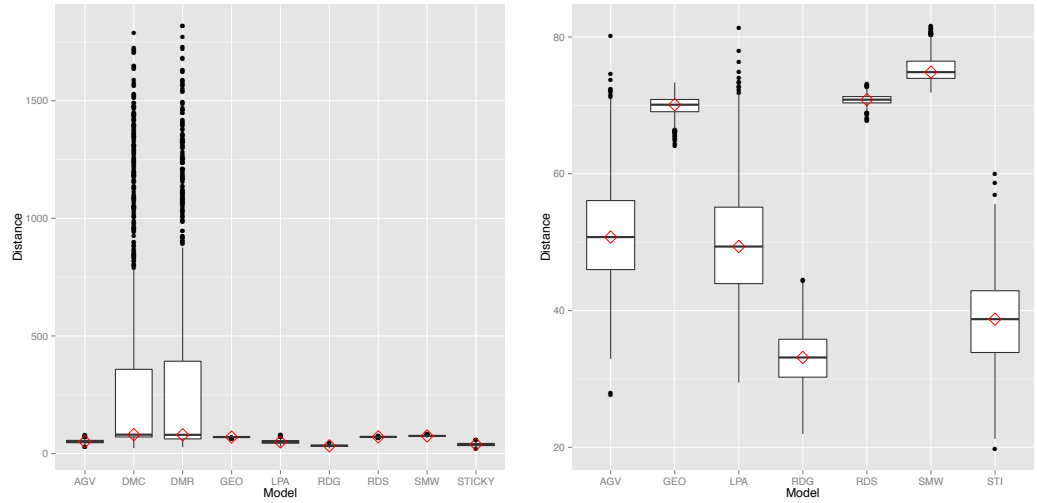
8.2.1 Degree Distribution Distance

The degree distribution distance identifies RDG as the best model for the *S. cerevisiae* PPI network. In Figure 8.1 we see two images of the comparisons. The figure on the left, Figure (8.1a) shows the results of all the comparisons while the figure on the right (Figure 8.1b) shows the comparisons with the DMC and DMR results not included. Due to the large spread seen in those two models it can be difficult to see any information about the others, thus the second picture provides a closer look.

We can see that AGV, LPA, RDG, and STI have larger spreads than GEO, RDS, and SMW. DMC and DMR are both very skewed towards higher distances. Since this method is not normalized it is hard to determine whether the differences in these distances are significant or not. We could be looking for the best of many bad choices or all the choices could be very good and differ by insignificant amounts.

We can use Bayes theorem to test the probabilities that the *S. cerevisiae* PPI network was classified as RDG given that it is RDG and given that it is not.

$$\begin{aligned}\Pr(\text{classified as RDG} \mid \text{RDG}) &= \frac{\Pr(\text{RDG} \mid \text{classified as RDG}) \cdot \Pr(\text{classified as RDG})}{\Pr(\text{RDG})} \\ &= \frac{\frac{100}{381} \cdot \frac{381}{900}}{\frac{1}{9}} \\ &= 1\end{aligned} \tag{8.10}$$



(a) (b)

Figure 8.1. ***S. cerevisiae* PPI network classification by the degree distribution distance.** Each figure shows the results of comparing the empirical *S. cerevisiae* PPI network against the 1000 model graphs of each of the nine types. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** The full results using the degree distribution distance. **(b):** Closeup, not including DMC, DMR.

Using Bayes, the probability that a model is classified as RDG given it is actually RDG is 100%.

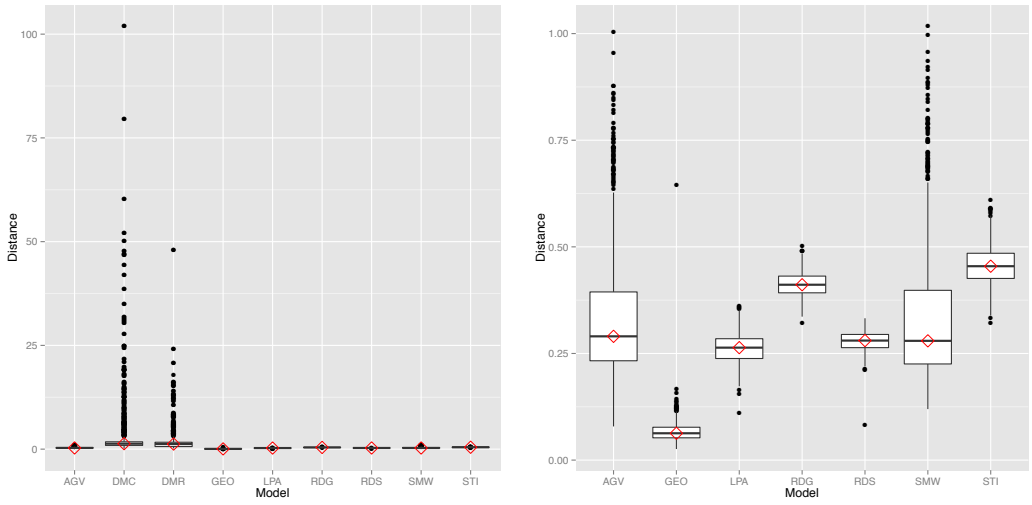
$$\begin{aligned}\Pr\left(\text{classified as RDG} \mid \text{NOT RDG}\right) &= \frac{\Pr(\text{NOT RDG} \mid \text{classified as RDG}) \cdot \Pr(\text{classified as RDG})}{\Pr(\text{NOT RDG})} \\ &= \frac{\frac{281}{381} \cdot \frac{381}{900}}{\frac{8}{9}} \\ &= 0.3513 \end{aligned} \tag{8.11}$$

Note that the probability of being RDG ($\frac{1}{9}$) and the probability of not being RDG ($\frac{8}{9}$) are empirically derived from our test comparisons.

We have a 35% chance that a model classified as RDG is actually not RDG, leaving us with some uncertainty as to whether the *S. cerevisiae* PPI network is actually RDG or not. In addition, the F-measure for RDG under DDD is only 0.4158. Therefore, combined, these values indicate that we can neither accept, nor reject, the choice of RDG for best fit under DDD; the statistics are inconclusive. Since the DDD is a novel algorithm, there are no previous results for comparison.

8.2.2 Characteristic Curve

The results for the characteristic curve look very similar to those of of the DDD (Figure 8.2, 8.1). The spreads of the DMC and DMR are once again very large, requiring a more zoomed in image. For the CC, GEO was designated as the best fit. It has a small IQR, and is slightly skewed towards higher values. LPA, RDG, RDS, and STI also all have small spreads. AGV and SMW are much more spread out with large whiskers towards the higher values. The method is not normalized so it is difficult to determine what constitutes a significant versus insignificant distance.



(a) (b)

Figure 8.2. ***S. cerevisiae* PPI network classification by the characteristic curve.** Each figure shows the results of comparing the empirical *S. cerevisiae* PPI network against the 1000 model graphs of each of the nine types. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** The full results using the characteristic curve. **(b):** Closeup, not including DMC, DMR.

Once again using Bayes Theorem, we can determine that the probability that the model is classified as GEO given that it really is GEO is 1 and the probability that it is classified in such a way given that it is not is 0.0363.

$$\begin{aligned}\Pr(\text{classified as GEO} \mid \text{GEO}) &= \frac{\Pr(\text{GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{GEO})} \\ &= \frac{\frac{100}{129} \cdot \frac{129}{900}}{\frac{1}{9}} \\ &= 1 \end{aligned} \tag{8.12}$$

$$\begin{aligned}\Pr\left(\text{classified as GEO} \mid \text{NOT GEO}\right) &= \frac{\Pr(\text{NOT GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{NOT GEO})} \\ &= \frac{\frac{29}{129} \cdot \frac{129}{900}}{\frac{8}{9}} \\ &= 0.0363 \end{aligned} \tag{8.13}$$

In addition, the F-measure for this choice is 0.8734. This is near the maximum value of one. Between this statistic, and the fact that a graph is incorrectly classified as GEO about 3.6% of time, one can confidently state that according the CC, the best model for the *S. cerevisiae* PPI network is the GEO model type.

In the paper introducing the CC, only DMC, DMR, and LPA graphs were compared to the three version of the *Drosophila melanogaster* PPI network (Su *et al.* , 2011). They were ranked such that DMC was first, DMR second, and LPA third. In this analysis, DMC was ranked ninth which is last place, DMR was eighth, and LPA was second. From Table 7.5 we cannot use Bayes theorem to calculate posterior classification probabilities because no graph was correctly, or incorrectly, classified as DMC.

8.2.3 Relative Graphlet Frequency

The final results for the relative graphlet frequency are significantly more condensed than either the CC or the DDD. DMC and DMR still have a much larger IQR than the other model types, but in this case the difference is not quite so large, Figure 9.3a. In addition, the spread of SMW is not much smaller. The remaining models all have the small spread of values like we have seen in previous plots.

The best fit here is once again RDG, but with GEO less than a tenth of a point larger in distance. Since this method is also unnormalized, making it difficult to determine the significance of a difference in a tenth of a point, we will consider the results for both of these model types.

$$\begin{aligned}\Pr\left(\text{classified as RDG} \mid \text{RDG}\right) &= \frac{\Pr(\text{RDG} \mid \text{classified as RDG}) \cdot \Pr(\text{classified as RDG})}{\Pr(\text{RDG})} \\ &= \frac{\frac{100}{172} \cdot \frac{172}{900}}{\frac{1}{9}} \\ &= 1 \end{aligned} \tag{8.14}$$

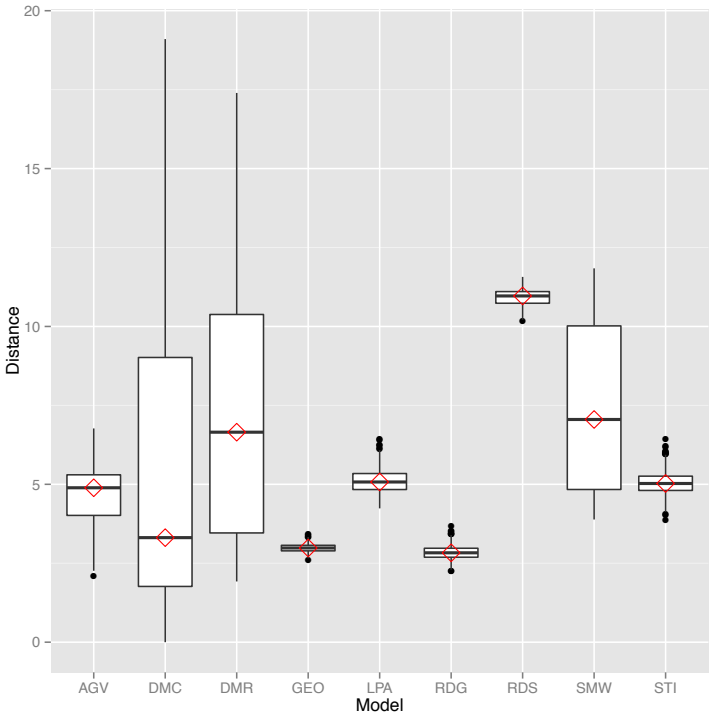


Figure 8.3. ***S. cerevisiae* PPI network classification by the relative graphlet frequency.** The figure shows the results of comparing the empirical *S. cerevisiae* PPI network against the 1000 model graphs of each of the nine types. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances.

$$\begin{aligned}\Pr(\text{classified as RDG} \mid \text{NOT RDG}) &= \frac{\Pr(\text{NOT RDG} \mid \text{classified as RDG}) \cdot \Pr(\text{classified as RDG})}{\Pr(\text{NOT RDG})} \\ &= \frac{\frac{72}{172} \cdot \frac{172}{900}}{\frac{8}{9}} \\ &= 0.09 \end{aligned} \tag{8.15}$$

$$\begin{aligned}\Pr(\text{classified as GEO} \mid \text{GEO}) &= \frac{\Pr(\text{GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{GEO})} \\ &= \frac{\frac{100}{127} \cdot \frac{127}{900}}{\frac{1}{9}} \\ &= 1 \end{aligned} \tag{8.16}$$

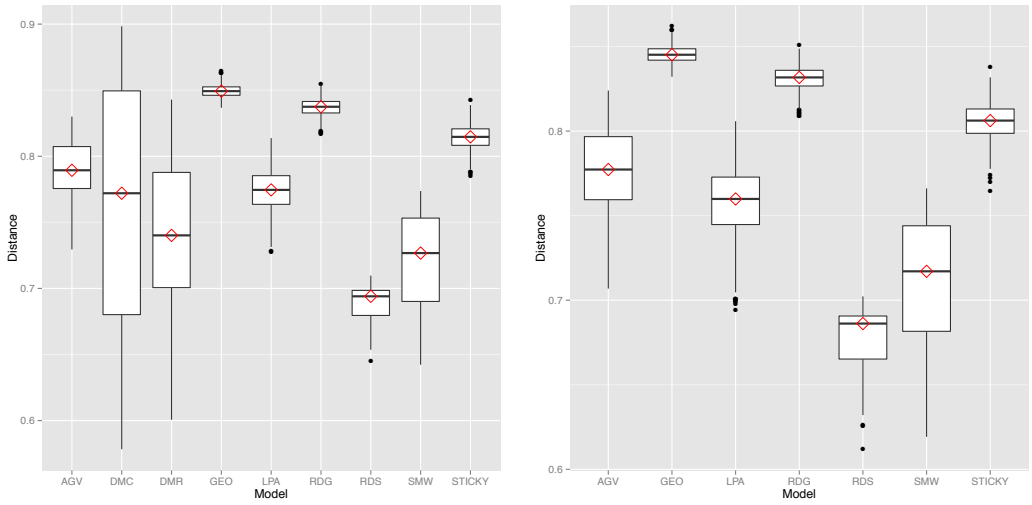
$$\begin{aligned}\Pr(\text{classified as GEO} \mid \text{NOT GEO}) &= \frac{\Pr(\text{NOT GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{NOT GEO})} \\ &= \frac{\frac{27}{127} \cdot \frac{127}{900}}{\frac{8}{9}} \\ &= 0.0338 \end{aligned} \tag{8.17}$$

For both RDG and GEO, we are 100% sure that if the model is in either class then it will be classified correctly. For RDG though, the probability that a model is incorrectly classified as RDG 0.09. This 9% chance of an incorrect model being classified as RDG is small. The F-measure for RDG is 0.7353. These values allow us to be reasonably confident in the choice of RDG as the best fit for the *S. cerevisiae* PPI network. In the case of GEO, the probability of a graph being incorrectly placed in that group is 0.0338. The F-measure is 0.885. Similar to the results of the CC, we can be a lot more confident in a result of GEO than in one of RDG. In the paper introducing relative graphlet frequency, GEO was also found to be the best fit for all of the PPI network (Przulj *et al.* , 2004).

8.2.4 Graphlet Degree Distribution

The Graphlet Degree Distribution uses an agreement to determine how alike two graphs are, as opposed to a distance. For this reason, when considering the images seen in Figures 8.4a, 8.4b it is important to remember that the best fit is the model with the highest score.

In both figures, we still see the large spread in results for DMC and DMR that was found for all the other methods. Similar to RGF, the overall difference in median results is small enough that all of the values can be seen comfortably on the same plot. The overall distance between the best and worst fit is only 0.1553 for GDD (A) and 0.1589 for GDD (G).



(a) (b)

Figure 8.4. **Comparison of *S. cerevisiae* PPI network classification by the graphlet degree distribution using arithmetic mean and using geometric mean.** Each figure shows the results of comparing the empirical *S. cerevisiae* PPI network against the 1000 model graphs of each of the nine types. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** Results for graphlet degree distribution using the arithmetic mean. **(b):** Results for graphlet degree distribution using the geometric mean.

For GDD (A) and GDD (G), the best fit is GEO. This was also the best fit found by the original paper (Przulj, 2007) for all of the fourteen PPI networks examined. In both instances of the GDD, we are 100% sure that if the *S. cerevisiae* PPI network is truly GEO, then it will be classified as such. Using the arithmetic mean, we are at a 20% risk of classifying a graph that is not actually GEO as GEO.

$$\begin{aligned}\Pr(\text{classified as GEO} \mid \text{GEO}) &= \frac{\Pr(\text{GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{GEO})} \\ &= \frac{\frac{100}{260} \cdot \frac{260}{900}}{\frac{1}{9}} \\ &= 1 \end{aligned} \tag{8.18}$$

$$\begin{aligned}\Pr\left(\text{classified as GEO} \mid \text{NOT GEO}\right) &= \frac{\Pr(\text{NOT GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{NOT GEO})} \\ &= \frac{\frac{160}{260} \cdot \frac{260}{900}}{\frac{8}{9}} \\ &= 0.2 \end{aligned} \tag{8.19}$$

This risk increases to 22.5% when the geometric mean is used.

$$\begin{aligned}\Pr\left(\text{classified as GEO} \mid \text{GEO}\right) &= \frac{\Pr(\text{GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{GEO})} \\ &= \frac{\frac{100}{280} \cdot \frac{280}{900}}{\frac{1}{9}} \\ &= 1 \end{aligned} \tag{8.20}$$

$$\begin{aligned}\Pr\left(\text{classified as GEO} \mid \text{NOT GEO}\right) &= \frac{\Pr(\text{NOT GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{NOT GEO})} \\ &= \frac{\frac{180}{280} \cdot \frac{260}{900}}{\frac{8}{9}} \\ &= 0.225 \end{aligned} \tag{8.21}$$

In addition to the high probabilities of incorrectly classifying a model as GEO using either GDD, the corresponding F-measures are 0.5556 and 0.5263 for the method using the arithmetic or geometric, respectively. These values indicate that the method correctly classifies models as GEO barely greater than 50% of the time. Therefore, we cannot, accept the results of the GDD.

8.2.5 Kendall's W Comparison of Ranking Lists

There are several things to note about the overall rankings provided by the five classifiers. First, examining the whole list, we have $W = 0.575$. This statistic is right in the middle of the range, and thus indicates that while the agreement across the classifiers is not perfect it is better than random. This is not entirely unexpected given that the classifiers all prioritize different model features. However, if we break the lists up into thirds, several similarities do appear. In the first third, GEO and RDG both appear in the top third in four out of the five methods. AGV and RDS never appear in the top third. AGV appears in the middle

third in each method. It is in fourth place for four methods fifth place for one (CC). Finally, DMR appears in the bottom third under each classifier. SMW appears in the bottom third in four of the five classifications and in the first third in one (CC). Only two models showed up in every third: DMC and STI. On the overall scale, there are not many similarities between the ranking orders of the five methods, however when the lists are separates into chunks, patterns do emerge.

8.3 Discussion

In this chapter, we determined the classifications of the *S. cerevisiae* PPI network by each of the five classifiers. The results were analyzed using Bayes theorem in an attempt to determine whether the results are reliable. We then compared the classification results from this dissertation to those found in the original papers, looking for the affect of using a minimal list of model graphs. Finally, we compared the overall rankings from the five classifiers to each other using Kendall's W. So, can we trust results obtained from any of these five methods? The resounding answer is an anticlimactic maybe.

In the case of DDD and both GDD, the probability that a model was incorrectly classified into the chosen model type is above 20%. This statistic negates the positive accuracy seen. It does not matter that the chance that a graph classified as GEO is actually GEO is 100% (or is RDG in the case of DDD), when there is such a large chance that a model is incorrectly placed into that category.

The CC and RGF had more reliable results. The chance that a graph was mistakenly classified as GEO by CC is only 3.6%. This is an acceptable margin of error. The relative graphlet frequency had two results whose distance only differed by a tenth of a point, RDG and GEO. The chance of misclassification in both these instances was 9% for RDG and 3.4% for GEO, both acceptable margins of error.

Finally, only two models were chosen to be the best fit for the *S. cerevisiae* PPI network. RDG and GEO. RDG was chosen by DDD and RGF. GEO was chosen by CC and both GDD. It was also a very close second for RGF. There are logical, and biological arguments for both model types being the best fit. For RDG, this is a growing graph, unlike GEO. Growing graphs have intrinsic properties, such as node age, that mimic PPI network

very well. However, in Chapter 4, we saw that RDG only matched 28% of the measures examined. One was a size measure (number of edges) and the remaining four were distance measures (diameter, radius, small-world properties, and scale-free property). None of the more complicated features seen in the centrality measures or connection measures were matched. This is to be expected because RDG is a random graph and those by definition lack complex features. RDG was also never chosen by the original classifier analyses to be the best fit.

GEO, on the other hand, was a popular choice by several of the classification mechanism. It was found the best fit by Przulj for both RGF and GDD. GEO also matched more features that RDG, 39%. It matched all of the size measures and then one each for distance, centrality and connection measures. The distance measure was scale-free property, centrality was degree centrality, and connection was average degree. Overall, it is not obvious if one of the choices is clearly the best fit for the *S. cerevisiae* PPI network at this point. Therefore, we proceed forward by editing two of the classification methods and presenting a novel one in the hopes that one of these will provide a concrete answer to the question: what model graph is the best fit for the *S. cerevisiae* PPI network?

Chapter 9

Relative Graphlet Frequency Error

Several methods to determine the growth mechanism of PPI networks were introduced in Chapter 5. Unfortunately, because of the complicated nature of these classifiers, errors do occur. One particular method that was found to have errors is the relative graphlet frequency (RGF) (Przulj *et al.* , 2004). This method uses the small 3- to 5-node subgraphs, or graphlets, to compare networks.

Due to the nature of the analyses performed and a desire for the output to be organized in a specific manner, it was easier to rewrite portions of the RGF algorithm. The original algorithm for the classifier can be found in the software package GraphCrunch 2 - version 2.1.1 (Kuchaiev *et al.* , 2011). GraphCrunch was used to calculate the numbers of graphlets, however distance calculations were rewritten in python code. This allowed for greater control over the form of the output, reduced redundancies in calculations, and led to the discovery of the error. In this chapter, we discuss the error and propose a solution to fix it. We will then present the results of the edited RGF, which will be referred to as the corrected relative graphlet frequency or RGF (C). These results will be compared to the results of the original RGF algorithm, presented in Chapters 6 and 7.

9.1 Formula Error

In her paper introducing the relative graphlet frequency, Pruzlj (Przulj *et al.* , 2004) defined the distance between graphlet frequencies as:

$$\mathcal{D}_{\mathcal{RGF}}(\mathcal{G}_1, \mathcal{G}_2) = \sum_{i=1}^{29} \left| F_i(\mathcal{G}_1) - F_i(\mathcal{G}_2) \right|, \tag{9.1}$$

where $F_i(\mathcal{G})$ is defined as:

$$F_i(\mathcal{G}) = -\log\left(\frac{N_i(\mathcal{G})}{T(\mathcal{G})}\right). \tag{9.2}$$

Interestingly, in the examination of the source code, it was discovered that the definition of $F_i(\mathcal{G})$ was inappropriately calculated. Instead of using Equation 9.3, $F_i(\mathcal{G})$ was defined as:

$$F_i(\mathcal{G}) = \frac{-\log\left(N_i(\mathcal{G})\right)}{\log\left(T(\mathcal{G})\right)}. \tag{9.3}$$

Clearly Equation 9.2 and Equation 9.3 are not equal. The former equation, Equation 9.2, was given in the paper where the concept was introduced (Przulj *et al.* , 2004) and is, presumably, the desired equation. An appropriate equality for Equation 9.2 is given in Equation 9.4.

$$F_i(\mathcal{G}) = \log\left(T(\mathcal{G})\right) - \log(N_i\left(\mathcal{G}\right)) \tag{9.4}$$

This error most likely stems from the dissertation of the developer of GraphCrunch (Kuchaiev *et al.* , 2011). In this document, $F_i(\mathcal{G})$ was defined using Equation 9.3.

While this is just a basic math error, there is one situation in which the use of the incorrect formula is truly detrimental. If there is only one graphlet in the network, then $T(\mathcal{G}) = 1$. This does not cause any mathematical issues if the correct form of $F_i(\mathcal{G})$ is used (Equation 9.4), however it results in an undefined value in Equation 9.3 since the denominator becomes zero:

$$\log\left(T(\mathcal{G})\right) = 0. \tag{9.5}$$

This situation was rare in these contexts, though it did appear in one of the 9000 model graphs. The graph with only one graphlet is a DMR model graph. The creators of GraphCrunch rectified this issue by setting $F_i(\mathcal{G}) = 0$ whenever $T(\mathcal{G}) = 1$. Using the correct formula, Equation 9.2 eliminates the need for this solution and also properly utilizes the desired logarithmic properties.

9.2 Methods

We rectify the use of the original, improper RGF algorithm by repeating the simulation comparisons performed in Chapter 7. Thus, the same 180 random test graphs created with

different probabilities were classified along with the 900 model test graphs, 100 from each of nine types. Results were analyzed using the same metrics: accuracy, F-measures, sensitivity, specificity, PPV, and NPV. Finally, the classification of the empirical *S. cerevisiae* PPI network was performed and the results interpreted using Bayes theorem. The results from both the original and corrected RGF are compared and contrasted.

It is necessary to show the results obtained using the incorrect formula (Chapter 7) in order to compare the results obtained in this dissertation with those obtained in the original papers (Przulj *et al.* , 2004). The purpose of repeating the analyses using the corrected formula is to determine the affect of the mathematical error on the results.

9.3 Results

9.3.1 Random Graph Classification

Both the original and corrected versions of the relative graphlet frequency classified all of the random graphs correctly.

9.3.2 Model Graph Classification

The difference in results for model graph classification can be seen in Table 9.1. The value in parentheses corresponds to the percent correctly classified using the original, incorrect, formula. The other value is the percent correctly classified by the corrected version of the RGF. The main difference occurs in the classification of AGV and DMC. The original RGF classified 13% of the AGV and none of the DMC graphs correctly. The corrected version classified none of the AGV and 12% of the DMC graphs correctly. The only other difference between the numbers of models of each type correctly classified is in SMW. The original RGF classified two correctly, while the corrected version only classified one. These changes result in the same overall classification accuracy for the two versions, 57%.

Despite the fact that the overall classification accuracies for the original and corrected RGF are the same, the two versions incorrectly classified graphs very differently from each other. Figure 9.1, shows how the model graphs are misclassified for AGV (a), DMC (b), DMR (c) and SMW (d). The other five model types were always classified correctly the the corrected RGF.

Table 9.1. Classification accuracy of both the original relative graphlet frequency and the corrected relative graphlet frequency.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
Actual Class	AGV	(13)	1	-	2	58 (68)	39 (19)	-	-	-
	DMC	(4)	12	-	10 (11)	(2)	48 (33)	23 (29)	2 (1)	3 (20)
	DMR	-	-	-	1 (1)	-	9 (16)	42 (48)	1 (35)	48
	GEO	-	-	-	100 (100)	-	-	-	-	-
	LPA	-	-	-	-	100 (100)	-	-	-	-
	RDG	-	-	-	-	-	100 (100)	-	-	-
	RDS	-	-	-	-	-	-	100 (100)	-	-
	SMW	(22)	-	-	20 (14)	1 (17)	25 (4)	36 (41)	1 (2)	17
	STI	-	-	-	-	-		-	-	100 (100)

Values in parentheses are from the original RGF classification where $F_i(\mathcal{G})$ was incorrectly calculated. Values outside of the parentheses are from the corrected RGF classification. The values presented are the percent of model graphs classified into each category by the given classifier.

For AGV, both methods misclassified models as LPA and RDG, however only the corrected version also used both DMC and GEO. Only the original RGF classified any AGV correctly.

DMC model misclassification is very spread out. Both the original and corrected RGF incorrectly classified DMC as GEO, RDG, RDS, SMW and STI. The original also misclassified DMC as AGV and LPA. Only the corrected version correctly classified any DMC models. The most common misclassification choice for DMC was AGV under the original RGF classifier and RDG under the corrected classifier.

DMR misclassification is not as widely spread as DMC. The original only misclassified DMR into four incorrect model types, GEO, RDG, RDS, and SMW. RDS was the most common incorrect choice. The corrected RGF added STI to the list of incorrect DMR classification choices. RDG was the common choice for this version.

The misclassifications of the SMW graphs is almost as spread out as the DMC classification. Both versions of RGF incorrectly classified SMW as GEO, LPA, RDG, and RDS, though only the original RGF misclassified SMW as AGV and only the corrected version classified it as STI.

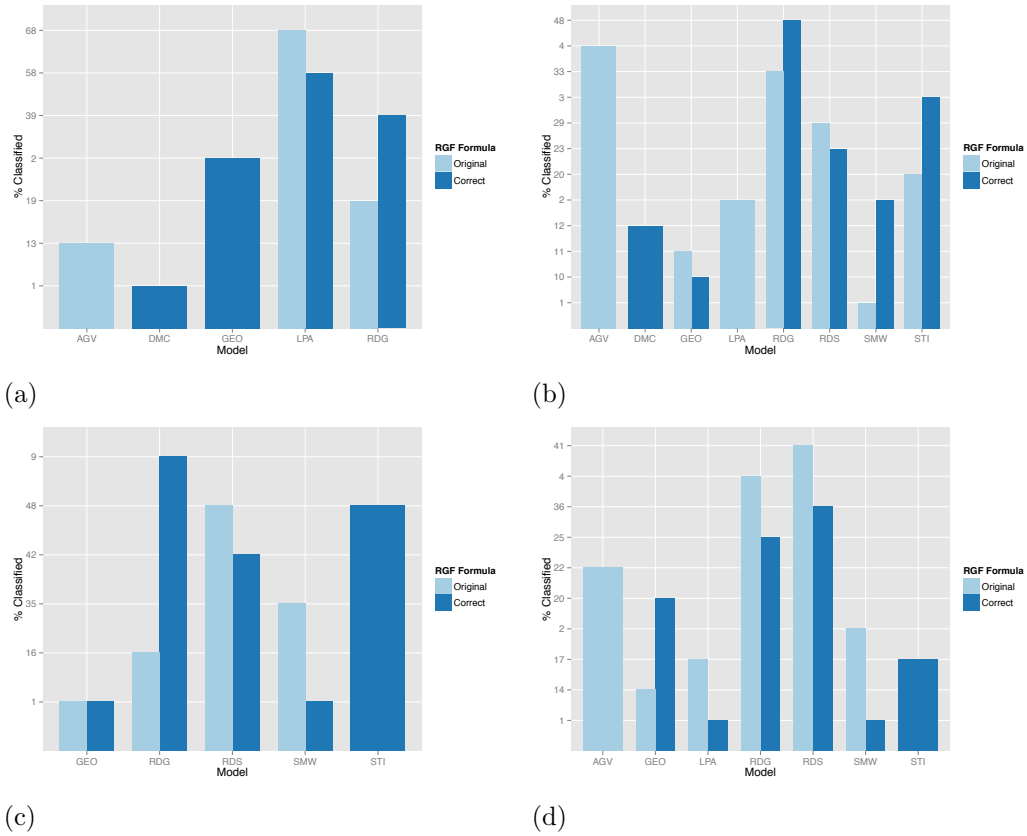
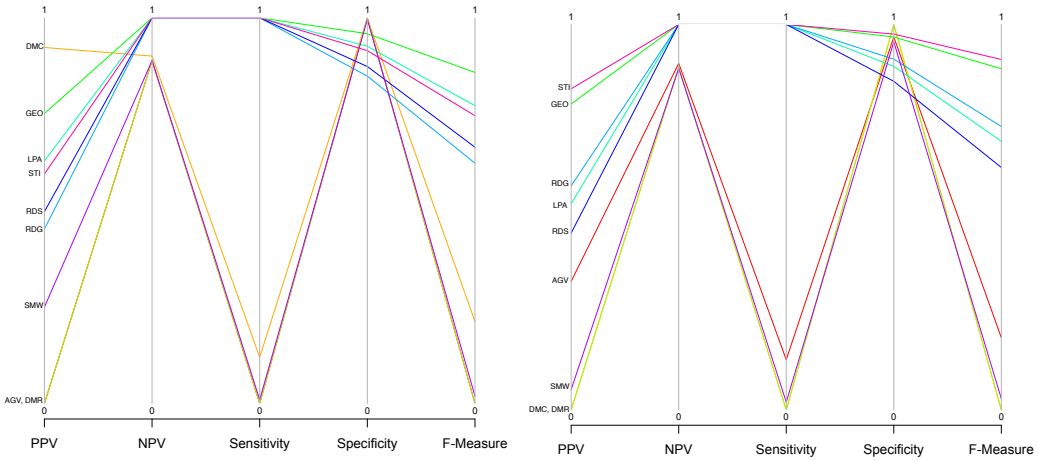


Figure 9.1. **Incorrect model graph classification by the original relative graphlet frequency and corrected relative graphlet frequency.** Each frame shows a comparison of the distribution of incorrectly classified model graphs by the original RGF and corrected RGF. The remaining five model graphs were never classified incorrectly and thus are not displayed. **(a):** Results for AGV. **(b):** Results for DMC. **(c):** Results for DMR. **(d):** Results for SMW.

It is interesting that despite having the same overall classification accuracy, there are some clear differences in how the models graphs are being assessed and categorized. This idea is further represented in Figure 9.2. This figure compares the analyses of performance for both the correct and original version of RGF. The pictures appear very similar, but based on the groups discussed in Chapter 7 the lines representing two model graphs change groups.These are AGV and DMC. In the original RGF, AGV can be classified in Group 4 and DMC as Group 2. This means that AGV was classified correctly with moderate accuracy and very few graphs were incorrectly classified as AGV. DMC, on the other hand, had low classification accuracy but was also an unpopular incorrect choice. In the corrected model, AGV is Group 2 and DMC is Group 3. Models in Group 3 have low classification

accuracy and are almost never chosen incorrectly.



(a) (b)

Figure 9.2. **Comparison of original and corrected relative graphlet frequency performance Statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 100 models of each of the nine model graph types. **(a):** This figure shows the corrected relative graphlet frequency; the version without the calculation error. **(b):** This figure shows the original relative graphlet frequency; the version found in GraphCrunch 2.

Table 9.2 provides a more precise look at the values seen in Figure 9.2. When compared to the statistical values of the original RGF (reproduced in Table 9.3), the average values for the corrected version are all larger. The differences, however, are all very small, with the exception of F-micro. The F-micro for the original RGF is 0.45 and 0.58 for the correct version (Tables 9.2, 9.3), thus we can conclude the corrected version of RGF is a better classifier than the original.

Table 9.2. Corrected relative graphlet frequency statistical analysis of performance.

Model	PPV	NPV	Sensitivity	Specificity	F-measure
AGV	0.0	0.8889	0.0	1.0	0.0
DMC	0.9231	0.9008	0.12	0.9988	0.2124
DMR	0.0	0.8889	0.0	1.0	0.0
GEO	0.7519	1.0	1.0	0.9588	0.8584
LPA	0.6289	1.0	1.0	0.9263	0.7722
RDG	0.4525	1.0	1.0	0.8488	0.6231
RDS	0.4975	1.0	1.0	0.8738	0.6645
SMW	0.25	0.8895	0.01	0.9963	0.0192
STI	0.5952	1.0	1.0	0.915	0.7463
Average	0.4555	0.952	0.57	0.8353	0.4329
Global	0.5706	0.9999	0.57	0.9464	0.5703

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the corrected RGF. (The corrected RGF is the edited version without the calculation error in $F_i(\mathcal{G})$.) Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

Table 9.3. Original relative graphlet frequency statistical analysis of performance.

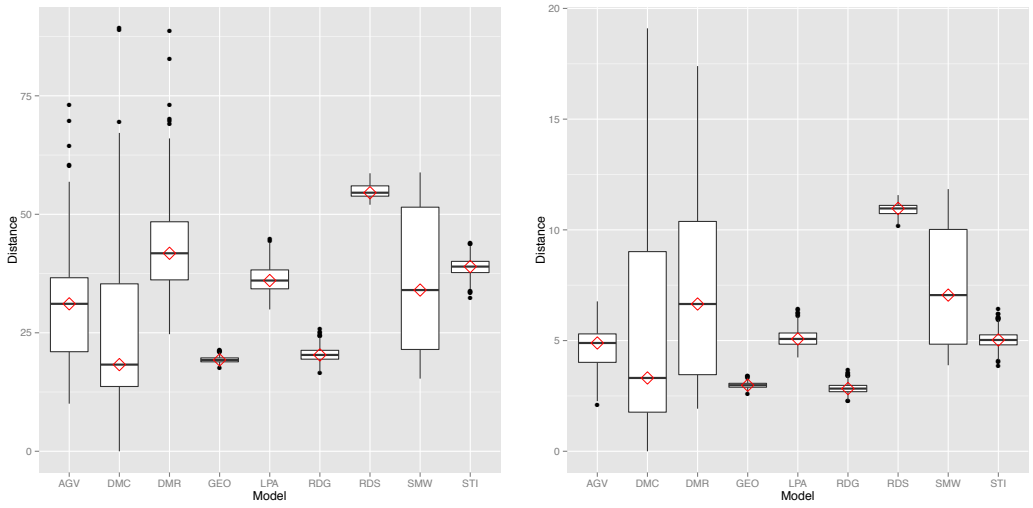
Model	PPV	NPV	Sensitivity	Specificity	F-measure
AGV	0.3333	0.899	0.13	0.9675	0.1871
DMC	0.0	0.8889	0.0	1.0	0.0
DMR	0.0	0.8889	0.0	1.0	0.0
GEO	0.7937	1.0	1.0	0.9675	0.8850
LPA	0.5348	1.0	1.0	0.8913	0.6969
RDG	0.5814	1.0	1.0	0.91	0.7353
RDS	0.4587	1.0	1.0	0.8525	0.6289
SMW	0.0526	0.8863	0.02	0.955	0.0290
STI	0.833	1.0	1.0	0.9750	0.9091
Average	0.3986	0.9514	0.5722	0.9465	0.4524
Global	0.5722	0.9999	0.5722	0.9465	0.5722

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the original RGF. (The original RGF is the version contained in GraphCrunch 2 with the calculation error in $F_i(\mathcal{G})$.) Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

9.3.3 Saccharomyces cerevisiae PPI Network Classification

We previously found that the original RGF chose RDG as the best fit for the *S. cerevisiae* PPI network (Chapter 8). There was a 100% chance that if the *S. cerevisiae* PPI network really is RDG, then it will be classified as RDG and a 9% chance that it still will be classified as RDG even if that is not the truth. Figure 9.3 shows a comparison of the *S. cerevisiae* PPI network classification between the corrected RGF (Figure 9.3a and the original RGF (Figure 9.3b). The positioning of the outcomes are very similar, though the scale is different. The figure on the left ranges from zero to one-hundred, while the one on the right

only goes to twenty. One interesting thing to note is that the corrected RGF is the only method in which the largest interquartile range is not seen in DMC and DMR, but in SMW.



(a) (b)

Figure 9.3. **Comparison of *S. cerevisiae* PPI network classification by original and corrected relative graphlet frequency.** Each figure shows the results of comparing the empirical *S. cerevisiae* PPI network against the 1000 model graphs of each of the nine types. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** The results using the corrected relative graphlet frequency. DMC is the best fit followed closely by GEO. **(b):** The results using the original relative graphlet frequency. RDG is declared the best fit, also followed closely by GEO.

The corrected RGF selected DMC as its best fit for the *S. cerevisiae* PPI network. Using Bayes theorem, we deduce that the probability that the model is DMC and is classified correctly is only 12% (Equation 9.6), however the probability that model is not DMC, but is classified as such is only 1% (Equation 9.7).

$$\begin{aligned}\Pr\left(\text{classified as DMC} \mid \text{DMC}\right) &= \frac{\Pr(\text{DMC} \mid \text{classified as DMC}) \cdot \Pr(\text{classified as DMC})}{\Pr(\text{DMC})} \\ &= \frac{\frac{12}{13} \cdot \frac{13}{900}}{\frac{1}{9}} \\ &= 0.12 \end{aligned} \tag{9.6}$$

$$\begin{aligned}\Pr\left(\text{classified as DMC} \mid \text{not DMC}\right) &= \frac{\Pr(\text{not DMC} \mid \text{classified as DMC}) \cdot \Pr(\text{classified as DMC})}{\Pr(\text{not DMC})} \\ &= \frac{\frac{1}{13} \cdot \frac{13}{900}}{\frac{8}{9}} \\ &= 0.01 \end{aligned} \tag{9.7}$$

It is clearly very rare for any model to be classified as DMC, only 1.4% were, and for any model classified as DMC, it is 8.5 times more likely that that model actually is a DMC model graph than anything other model.

Table 9.4 shows the rankings of all nine model graphs by both the correct and original RGF. The list have a Kendall's W of 0.908 and are not statistically significantly different at a significance level of 0.05. Between the two lists, four of the models appear in the same position in both lists. The top three model types in both lists are the same. If we continue to break the list into thirds, we see that only two of the graph switched thirds, STI and SMW. SMW is in 8^{th} position, or last third, when classified by the original RGF and moved into the second third when classified by the corrected RGF. STI completed the opposite trip, moving from 5^{th} position, in the middle third, to the last third.

Table 9.4. Ordered rankings of the model graphs based on fit for *S. cerevisiae* PPI network using the corrected and original relative graphlet frequency.

	Corrected RGF	Original Formula
1	**DMC**	**RDG**
2	GEO	GEO
3	**RDG**	**DMC**
4	AGV	AGV
5	**SMW**	**STI**
6	LPA	LPA
7	**STI**	**DMR**
8	**DMR**	**SMW**
9	RDS	RDS

Rankings of model graphs are determined based on the median distance of the *S. cerevisiae* PPI network to each model graphs of the given type. The median smallest distance is ranked first. Models in bold show up in different places across the two lists. Items not in bold do not change position when the corrected relative graphlet frequency is used in place of the original.

9.4 Conclusions

The program created to run the relative graphlet frequency algorithm contained a mathematical error in its calculations. This error was discovered when parts of the program rewritten in order to increase program runtime effficiency. In this chapter, we fixed the problem and repeated the analyses performed in Chapter 7. We then compared these results to the results of the original analyses to determine the effect of using an erroneous formula.

The differences in classification using the original and corrected formulas are minimal. Both performed perfectly on the random graph classification. Slight changes occurred when the method was tested on its ability to correctly classify the model graphs. The overall classification accuracy remained the same, but thirteen out of 9000 model graphs changed from correctly classified to incorrectly classified, or vice versa (Table 9.2). This, however, corresponds to only 0.1% of the graphs.

The changes in groups for two of the model graphs, AGV and DMC, occurs because of the way the corrected RGF analyzes graphs. For AGV, none of these model graphs were classified correctly and no model graphs were incorrectly classified as AGV. This is a change for thirteen models classified correctly using the original RGF and 26 model graphs being incorrectly classified as AGV. Essentially the opposite change in classification occurred for DMC resulting in the change in groups. Using the original RGF, no DMC model graphs were classified correctly and nothing was incorrectly classified as DMC. Under the classification of the corrected RGF, twelve graphs were correctly classified as DMC and one was incorrectly classified.

Since the two versions classified graphs slightly differently, based on the fact that graphs were not classified incorrectly into the exact same categories, we can speculate that the change in formula is causing different characteristics within the model graphs themselves to be picked up on. However, under the circumstances, these differences are not substantial enough to promote vast differences in the results. This effect could be investigated further by determining the exact classification of each individual graph and looking to see how many change, but this investigation is not necessary in this instance be the overall impact is so minimal.

A change in classification did occur when the empirical *S. cerevisiae* PPI network was classified. In the original relative graphlet frequency, RDG was the best fit. However, when the corrected relative graphlet frequency was utilized, DMC was the best fit. Even though this difference occurred, the overall rankings of model graphs between the two groups is not statistically significantly different based on Kendall's W (Table 9.4). This signifies that the use of the incorrect formula may not have been entirely detrimental to the classification process.

Overall, the results between the original and corrected RGF did not differ drastically in terms of random graph, model graph, or *S. cerevisiae* PPI network classification. This is most likely to due the underlying reason for using the log in the first place. According to Przulj, the log is used because "frequencies of different graphlets can differ by several orders of magnitude" and this prevents frequently seen graphlets from dominating the distance (Przulj *et al.* , 2004). If none of the graphlets differ by several orders of magnitude, then the transformation whether applied correctly (corrected RGF) or incorrectly (original RGF) should not have much effect on the calculations. This might not be true in all situations and with different graphs used for comparison, results have the potential to be largely varied. Therefore, we conclude that the correct formula should be used to preserve mathematical integrity, even though the effect on the overall classification analyses is essentially negligible.

Chapter 10

Reformulations of the Graphlet Degree Distribution

The graphlet degree distribution classification algorithm is a complicated procedure. Since its inception, it has been the subject of critiques (Hayes *et al.* , 2015; Pržulj, 2010). Similar to the relative graphlet frequency, the computational burden was eased by reproducing pieces of the algorithm found in GraphCrunch 2 in original python code. Also similar to the relative graphlet frequency, the rewriting process illuminated several idiosyncracies of the method. Delving deeper into the source code, as well as reviewing the literature, simply showed that these idosyncracies were either ignored entirely or patched with bandaids. In this chapter, we discuss the weaknesses that were identified in the graphlet degree distribution. We then propose three alternative methods, all of which maintain the same idea as the original GDD, but do not possess the idiosyncracies discussed, and compare the methods to determine which is the best reformulation.

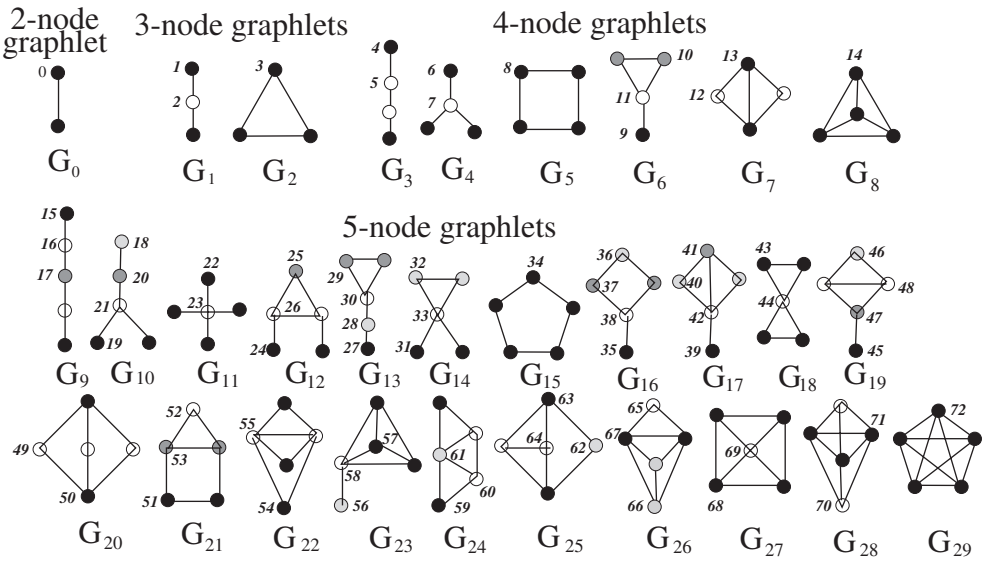


Figure 10.1. **Display of the 73 automorphism orbits (Przulj, 2007)-Figure 1.** Automorphism orbits are unique nodes position within each graphlet. They are differentiated by different color nodes in the image.

10.1 Graphlet Degree Distribution Issues

Before examining the problems found in the GDD algorithm it is useful to get a better idea of what exactly the algorithm is calculating. Consider the graphlet seen in Fig. 10.2. Based on Fig. 5.1, this is Graphlet 6 and it contain automorphism orbits 10, 11, and 12 (Fig. 5.2). Graphlet 6 is also referred to as the flower graphlet (Pržulj & Higham, 2006). It is composed of Graphlet 1 (path of length two) and Graphlet 2 (triangle). These are made up of automorphism orbits 1 and 2 for the path and 3 for the triangle. This results in the graphlet degree distribution seen in Table 10.1.

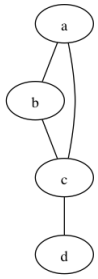


Figure 10.2. **Graphlet #6 (flower).**

Table 10.1. Graphlet degree distribution for Figure 10.2.

Automorphism Orbit (j)	k	$d_{\mathcal{G}}^{j}(k)$
1	1	2
1	2	1
2	2	1
3	1	3
9	1	1
10	1	2
11	1	1

This table shows the graphlet degree distribution for the flower graphlet (#6). The first column lists the automorphism corresponding to Figure 10.1. The second column, k, is the number of times a node acts as the given automorphism orbit. The third column, $d_{\mathcal{G}}^{j}(k)$, is the number of nodes that act as automorphism orbit j a total of k times.

Consider the first two lines of Table 10.1. These both refer to automorphism orbit $j = 1$, which is the end of node of the path of length two. The middle column refers to the number paths in which nodes of automorphism orbit 1 take part. The last column, $d_{\mathcal{G}}^{j}(k)$, is the number of nodes that take part k times. A far easier interpretation of the graphlet degree distribution is that there are two nodes ('a' and 'b') that participate as the end

node of a path of length two once. The second line indicates that there is one node that participates as the end node of a path of length two twice (node 'd'). The third line now addresses the center node of a path of length two. There is one node that participates as the center node twice (node 'c'). The third line now moves onto automorphism orbit 3, any corner of a triangle. There are three nodes ('a', 'b', and 'c') that each participate in one triangle. Finally, the last three lines address the full graphlet. Automorphism orbit 9 is the end of the path of length two off of the triangle. Only node 'd' meets this description and it does so in only one flower. Automorphism orbit 10 refers to the corners of the triangle with degree two, the ones not participating in the path. There are two nodes ('a' and 'b') that each participate in one flower. Finally, the last line refers to the corner of the triangle that does participate in the path, thus having a degree of three. There is only one node ('c') that participates in one flower.

Recall, the distance for a specific graphlet degree distribution is given as

$$\mathcal{D}^j(\mathcal{G}_1, \mathcal{G}_2) = \frac{1}{\sqrt{2}} \left(\sum_{k=1}^{\infty} \left[N^j_{\mathcal{G}_1}(k) - N^j_{\mathcal{G}_2}(k) \right]^2 \right)^{1/2} \tag{10.1}$$

where j is any automorphism orbit, $j \in \{1, \ldots, 72\}$. The agreement for the graphlet degree distribution of j is calculated by

$$A^j(\mathcal{G}_1, \mathcal{G}_2) = 1 - D^j(\mathcal{G}_1, \mathcal{G}_2) \tag{10.2}$$

and the average can be represented by either the arithmetic or geometric mean.

10.1.1 Geometric Mean

The most basic issue that we discovered pertaining to the graphlet degree distribution method arises from the use of the geometric mean, given in Equation 10.3. In this formula the agreement from each automorphism orbit is multiplied. The closer the overall agreement

is to one, the better the agreement is between the two networks.

$$A_{geo}(\mathcal{G}_1, \mathcal{G}_2) = \left(\frac{1}{73} \prod_{j=0}^{72} A^j(\mathcal{G}_1, \mathcal{G}_2) \right)^{1/73} \tag{10.3}$$

The issue with this formula comes into play when we achieve maximum disagreement at any single automorphism orbit. Maximimum disagreement can also be thought of as having a distance of one, the maximum distance allowed. If $A^j(\mathcal{G}_1, \mathcal{G}_2) = 0$ for any $j \in \{0, \ldots, 72\}$, then the overall agreement calculated using the geometric mean, $A_{geo}(\mathcal{G}_1, \mathcal{G}_2)$, is also equal to zero. Thus even if there is perfect agreement at every other automorphism orbit, a single case of complete disagreement has unjustified influence on the resulting value. Example 1 shows one potential situation.

Example 1 Let there be two networks, $\mathcal{G}_1$ and $\mathcal{G}_2$, such that for any automorphism orbit, j', the graphlet degree distribution is

$$d^{j'}_{\mathcal{G}_1}(1) = 0 \qquad d^{j'}_{\mathcal{G}_2}(1) = 2 \tag{10.4}$$

$$d^{j'}_{\mathcal{G}_1}(2) = 1 \qquad d^{j'}_{\mathcal{G}_2}(2) = 0. \tag{10.5}$$

Then the corresponding scaled values, $S^j_{\mathcal{G}}(k) = d^j_{\mathcal{G}}(k)/k$, become

$$S^{j'}_{\mathcal{G}_1}(1) = 0 \qquad S^{j'}_{\mathcal{G}_2}(1) = 2 \tag{10.6}$$

$$S^{j'}_{\mathcal{G}_1}(2) = 1/2 \qquad S^{j'}_{\mathcal{G}_2}(2) = 0 \tag{10.7}$$

with normalized values of

$$N^{j'}_{\mathcal{G}_1}(1) = 0 \qquad N^{j'}_{\mathcal{G}_2}(1) = 1 \tag{10.8}$$

$$N^{j'}_{\mathcal{G}_1}(2) = 1 \qquad N^{j'}_{\mathcal{G}_2}(2) = 0. \tag{10.9}$$

Using Equation 10.1 to calculate the distance,

$$\begin{aligned}
\mathcal{D}^{j'}(\mathcal{G}_1, \mathcal{G}_2) &= \frac{1}{\sqrt{2}} \left(\sum_{k=1}^{\infty} \left[N_{\mathcal{G}_1}^{j'}(k) - N_{\mathcal{G}_2}^{j'}(k) \right]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} \left(\left[N_{\mathcal{G}_1}^{j'}(1) - N_{\mathcal{G}_2}^{j'}(1) \right]^2 + \left[N_{\mathcal{G}_1}^{j'}(2) - N_{\mathcal{G}_2}^{j'}(2) \right]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} \left([0-1]^2 + [1-0]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} (2)^{1/2} = 1 \\
\mathcal{A}^{j'}(\mathcal{G}_1, \mathcal{G}_2) &= 0 \qquad (10.10)
\end{aligned}$$

This results in a GDD agreement value of zero, perfect disagreement, at automorphism orbit j'. When $j \neq j'$, $j \in \{0, \ldots, 72\}$ let $\mathcal{G}_1$ and $\mathcal{G}_2$ have identical distributions. This leads to a distance of zero and an agreement of one. When the geometric mean is calculated in this situation, the maximum disagreement seen at j', Line 10.10, overwhelms the perfect agreement seen at every other automorphism orbit. This results in an overall agreement value of zero.

Compare this example to one of two networks showing high levels of disagreement at every automorphism orbit, but never quite reaching maximal disagreement. These networks would have a higher agreement value than seen in the example when calculated with the geometric mean. In addition, the results calculated using the geometric mean, as compared to the arithmetic mean, are not congruent. This can be seen in Table 10.2. The arithmetic mean correctly classified 68 percent of the networks. The geometric mean only classified 60 percent correctly. Only one network model, SMW, had more networks classified correctly using the geometric mean as opposed to the arithmetic mean. Therefore, the arithmetic mean is more accurate in classifying the model graphs than the geometric mean.

Table 10.2. Comparison of arithmetic and geometric mean graphlet degree distribution classification results.

	Classification Accuracy (% Correct)	
Network Type	GDD (A)	GDD (G)
AGV	60	36
GEO	100	94
DMC	0	10
DMR	10	0
LPA	100	71
RDG	100	100
RDS	100	100
SMW	47	49
STI	100	97
Average	**68**	**60**

GDD (A) is the graphlet degree distribution with arithmetic mean. GDD (G) is the graphlet degree distribution with the geometric mean. The value presented in the percentage of graphs classified accurately for each model type.

10.1.2 Contradictory Outcomes

In addition to the problem with the use of the geometric mean, there are a few additional situations where the agreement calculated by the GDD does not result in a justifiable outcome, consider Example 2.

Example 2 Once again, let there be two networks, $\mathcal{G}_1$ and $\mathcal{G}_2$, such that for any automorphism orbit, j', the graphlet degree distribution is

$$d_{\mathcal{G}_1}^{j'}(1) = 1 \qquad d_{\mathcal{G}_2}^{j'}(1) = 0. \tag{10.11}$$

Then the corresponding scaled values become

$$S_{\mathcal{G}_1}^{j'}(1) = 1 \qquad S_{\mathcal{G}_2}^{j'}(1) = 0 \tag{10.12}$$

$$T_{\mathcal{G}_1}^{j'} = 1 \qquad T_{\mathcal{G}_2}^{j'} = 0 \tag{10.13}$$

with normalized values of

$$N_{\mathcal{G}_1}^{j'}(1) = 1 \qquad N_{\mathcal{G}_2}^{j'}(1) = 0. \tag{10.14}$$

The normalized value for $\mathcal{G}_2$ seen at the end of Line 10.14 results in an undefined value. This is simply set to zero for ease of calculation. Using

Equation 10.1 to calculate the distance,

$$\begin{aligned}
\mathcal{D}^{j'}(\mathcal{G}_1, \mathcal{G}_2) &= \frac{1}{\sqrt{2}} \left(\sum_{k=1}^{\infty} \left[N_{\mathcal{G}_1}^{j'}(k) - N_{\mathcal{G}_2}^{j'}(k) \right]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} \left(\left[N_{\mathcal{G}_1}^{j'}(1) - N_{\mathcal{G}_2}^{j'}(1) \right]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} \left([1-0]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} (1)^{1/2} \\
&= 0.7071 \\
\mathcal{A}^{j'}(\mathcal{G}_1, \mathcal{G}_2) &= 1 - 0.7071 \\
&= 0.2929 \qquad (10.15)
\end{aligned}$$

This results in a low GDD agreement value of 0.2929 (Line 10.15).

Compare the agreement values for the single automorphism orbit j' seen in Examples 1 and 2. In Example 1, we see an agreement value of zero, i.e. perfect disagreement. In Example 2 the level of agreement (0.2929) is still quite low, but it is larger than zero (Line 10.15). This implies that by the GDD metric, the networks seen in Example 1 are more different at that single automorphism orbit than the networks seen in Example 2. This view is neither obvious nor indisputable. In fact it could easily be argued that the exact opposite is true; the networks from Example 2 are more different than the networks from Example 1 at j' because only one of the networks has any nodes at automorphism orbit j' in the latter example. This issue points to some deeper inconsistencies within the overall design of the distance, and corresponding agreement, metric. It also indicates that the logical definition of distance may not be the same as this metric's mathematical definition.

A visual example of the aforementioned issue can be seen in Figures 10.3a, 10.3b, 10.3c. The situation depicted in the figures shows the agreement of two networks at a single automorphism network, j'. The two networks compared begin with the same distribution as seen in Example 1: $\mathcal{G}_1$ has one node participating as automorphism j' twice and $\mathcal{G}_2$ has two nodes participating as j' once (Table 10.3).

Table 10.3. Generic graphlet degree distribution corresponding to Figure 10.3.

k	$d_{\mathcal{G}_1}^{j'}(k)$	$d_{\mathcal{G}_2}^{j'}(k)$
1	0	2
2	1	0
3	0	n

This table shows the graphlet degree distribution of two generic graphs at any automorphism orbit. The first column, k, is the number of times a node acts as the given automorphism orbit. The second and third columns, $d_{\mathcal{G}_1}^{j'}(k)$ and $d_{\mathcal{G}_2}^{j'}(k)$, are the numbers of nodes that act as the given automorphism a total of k times. These two columns represent distributions for two unique graphs, $\mathcal{G}_1$ and $\mathcal{G}_2$.

The x-axis in each subplot of Figure 10.3 represents the number of nodes (n) from $\mathcal{G}_2$ that participate as j' three times. The other network, $\mathcal{G}_1$, remains at a constant distribution and has no nodes acting as j' three times. The three different figures in Figure 10.3 show the same situation over different domains. Figure 10.3a displays the agreement on values of $d_{\mathcal{G}_1}^{j'}(3) \in [0, 10]$, Figure 10.3b goes up to 100, and Figure 10.3c goes up to 250. The different ranges are utilized to highlight different aspects of the plots.

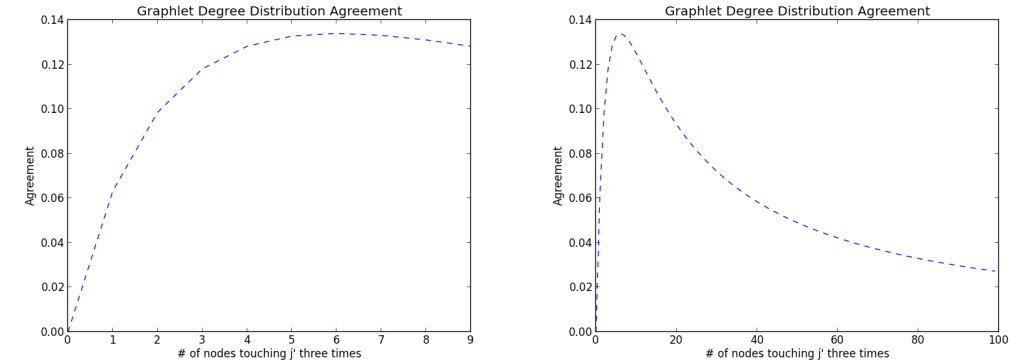
Several interesting things occur as the number of nodes touching j' three times increases. First, Figure 10.3a shows that the minimum agreement of zero is achieved when $\mathcal{G}_2$ does not have any nodes that touch j' three times, as in Example 2. The agreement quickly increases to its maximum value of 0.1340, which occurs when approximately six nodes touch j' three times (Figure 10.3b). From there the agreement begins to decrease (Figure 10.3c) and as:

$$d_{\mathcal{G}_2}^{j'}(3) \to \infty, \tag{10.16}$$

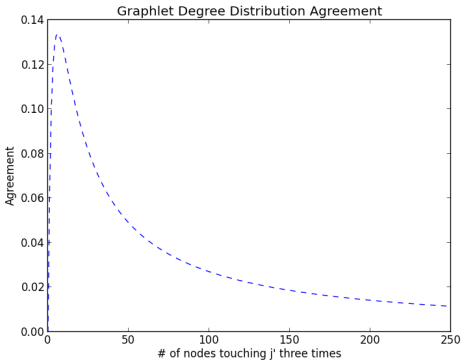
the agreement:

$$\mathcal{A}^{j'}(\mathcal{G}_1, \mathcal{G}_2) \to 0. \tag{10.17}$$

The plots in Figure 10.3 show that the minimum agreement is achieved when neither network has any nodes touching j' three times which is counterintuitive. Typically, one would expect a larger variation in the graphlet degree distribution to imply a larger distance and thus a smaller agreement. This is not the case with this metric. Another



(a) (b)



(c)

Figure 10.3. **GDD agreement over different domains at a single automorphism orbit as $d^{j'}_{\mathcal{G}_2}(3) \to \infty$.** The different domains each show distinct features of the agreement. The agreement shown corresponds to the graphlet degree distribution seen in Table 10.3. **(a):** The x-axis shows only $[0, 9]$. The values peak at approximately $x = 6$. **(b):** The x-axis shows $[0, 100]$. There is a smooth drop-off after the peak. **(c):** The x-axis shows $[0, 250]$. The line begins to near zero, however this is an asymptote. Zero is never reach again, it only happens at $x = 0$.

interesting feature is the dramatic increase to the maximum, which once again occurs at $x = 6$. In terms of biology or graph theory, there does not appear to be any significance of this number of nodes touching an automorphism orbit three times. Thus, this must be an artifact of the metric. Logic would dictate a constant decrease in agreement as $d^{j'}_{\mathcal{G}_2}(3) \to \infty$. The increase occurs because the addition of a new degree to the distribution increases $T^{j'}_{\mathcal{G}_2}$ and decreases the impact of each degree on the overall distance. An example of this can be seen below in Example 3.

Example 3 Assume that $\mathcal{G}_1$ and $\mathcal{G}_2$ have the exact same distributions for automorphism orbit j' as illustrated in Ex. 1. Then add to $\mathcal{G}_2$ one node that touches j' three times such that

$$d^{j'}_{\mathcal{G}_2}(3) = 1 \tag{10.18}$$
$$S^{j'}_{\mathcal{G}_2}(3) = 1/3 \tag{10.19}$$
$$T^{j'}_{\mathcal{G}_2} = 7/3 \tag{10.20}$$
$$N^{j'}_{\mathcal{G}_2}(1) = \frac{2}{7/3} = 6/7 \tag{10.21}$$
$$N^{j'}_{\mathcal{G}_2}(2) = \frac{0}{7/3} = 0 \tag{10.22}$$
$$N^{j'}_{\mathcal{G}_2}(3) = \frac{1/3}{7/3} = 1/7. \tag{10.23}$$

Then the distance can be calculated by

$$\begin{aligned}
\mathcal{D}^{j'}(\mathcal{G}_1, \mathcal{G}_2) &= \frac{1}{\sqrt{2}} \left(\sum_{k=1}^{\infty} \left[N^{j'}_{\mathcal{G}_1}(k) - N^{j'}_{\mathcal{G}_2}(k) \right]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} \left(\left[N^{j'}_{\mathcal{G}_1}(1) - N^{j'}_{\mathcal{G}_2}(1) \right]^2 + \left[N^{j'}_{\mathcal{G}_1}(2) - N^{j'}_{\mathcal{G}_2}(2) \right]^2 + \left[N^{j'}_{\mathcal{G}_1}(3) - N^{j'}_{\mathcal{G}_2}(3) \right]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} \left([0 - 6/7]^2 + [1 - 0]^2 + [0 - 1/7]^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} \left(36/49 + 1 + 1/49 \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} \left(1.76 \right)^{1/2} \\
&= 0.9381 \\
\mathcal{A}^{j'}(\mathcal{G}_1, \mathcal{G}_2) &= 1 - 0.9381 \\
&= 0.0619
\end{aligned} \tag{10.24}$$

The agreement value achieved here is greater than the agreement achieved in Example 1.

As previously stated, in the above example the addition of a new degree to the distribution increases $T^{j'}_{\mathcal{G}_2}$ thereby decreasing the impact of each degree on the total singular automorphism agreement. However, it is not clear that the networks in this scenario are less different than those seen in the previous example despite the fact that mathematically the statement is true. In fact, the more nodes that $\mathcal{G}_2$ has that touch j' three times, the larger the distance one would expect. Fortunately, this begins to occur after the minimum value. As $d^{j'}_{\mathcal{G}_1}(3) \to \infty$ the distance begins to approach its maximum value. The implication of this is that one of the degree distributions must approach zero while the other must approach one as the number of nodes increases. This is the only way to achieve the maximum distance of one (Example 1) using the GDD distance equation.

Example 4 Now consider the situation where networks $\mathcal{G}_1$ and $\mathcal{G}_2$ have the exact same distributions for automorphism orbit j' as illustrated in the previous example, but this time the number of nodes in $\mathcal{G}_2$ that touches j' three times is n such that

$$d^{j'}_{\mathcal{G}_2}(3) = n \tag{10.25}$$

$$S^{j'}_{\mathcal{G}_2}(3) = n/3 \tag{10.26}$$

$$T^{j'}_{\mathcal{G}_2} = (6+n)/3 \tag{10.27}$$

$$N^{j'}_{\mathcal{G}_2}(1) = \frac{1/2}{(6+n)/3} = \frac{3}{12+2n} \tag{10.28}$$

$$N^{j'}_{\mathcal{G}_2}(2) = \frac{0}{(6+n)/3} = 0 \tag{10.29}$$

$$N^{j'}_{\mathcal{G}_2}(3) = \frac{n/3}{(6+n)/3} = \frac{n}{6+n}. \tag{10.30}$$

If $n \to \infty$, then $N^{j'}_{\mathcal{G}_2}(1) \to 0$ while $N^{j'}_{\mathcal{G}_2}(3) \to 1$. Thus this scenario approaches Example 1 resulting in an agreement of zero.

In Example 4, the large number of nodes acting as j' three times effectively dominates the distance. This is one of the few expected outcomes of this method and the overall result can definitely be considered logical. Therefore, this method is far more sound when

when there are large numbers of nodes touching each automorphism orbit and mainly breaks down when the number of nodes at each graphlet degree is less than six.

10.1.3 Scaling and Normalization

The final statements regarding the GDD algorithm are merely critiques of the method. They do not represent mathematical issues or anything else that requires changing in order for the method to be mathematically sound, unlike the previously mentioned issues. The first of two final critiques involves the scaling step seen in Equation 10.31:

$$S^j_{\mathcal{G}}(k) = \frac{d^j_{\mathcal{G}}(k)}{k}. \tag{10.31}$$

In this step, the number of nodes acting as automorphism orbit j' a total of k times is scaled by k. Thus as $k \to \infty$,

$$S^{j'}_{\mathcal{G}}(k) \to 0. \tag{10.32}$$

Another way to look at Equation 10.32 is that the more times a node is acting as a particular automorphism orbit, the less of an impact it will have on the overall result. This results in a gradual leveling off in the agreement as k increases (Figure 10.4).

Przulj explained the reasoning behind this as an effort to "decrease the contribution of larger degrees in a GDD" (Przulj, 2007). The desire to decrease this contribution originates from her work with yeast PPI network and the finding that most counts above $k = 20$ were zero. This created a lot of noise in her data. Instead of applying a broad-band filter, Przulj chose to keep the data as was, but decrease the contribution of higher k. This approach is a problem-specific solution that reduces the generalizability of the method. It also negates the effect of high degree distributions that are not noise. Its ability to be applied to networks with high graphlet degree distributions is compromised.

Lastly, normalization of the GDD makes the juxtaposition of different graph comparisons easier to interpret. However, the chosen normalization allows for the ends of the range, zero and one, to be actually reachable. In the discussion of agreement, a value of

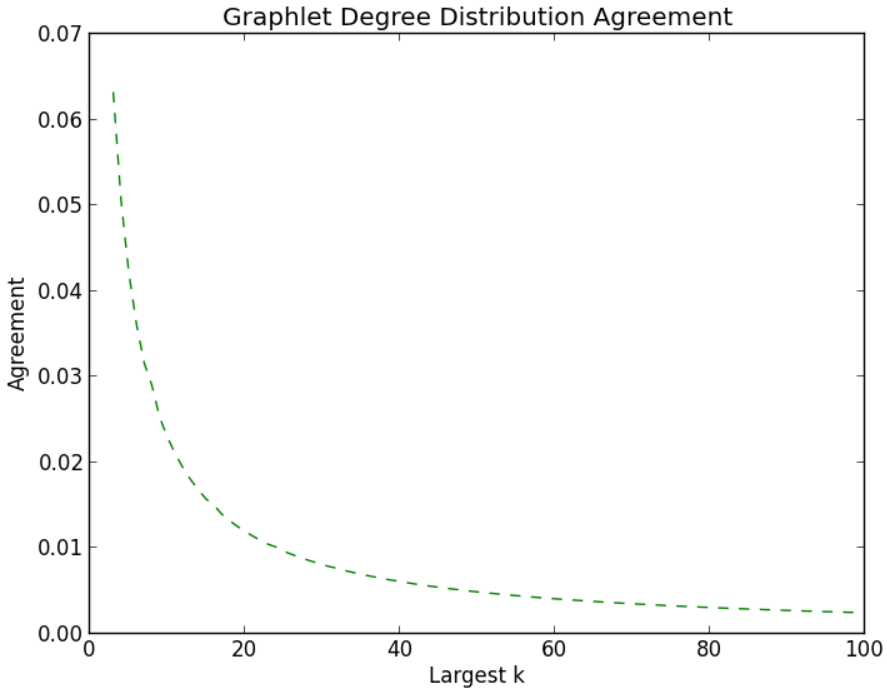


Figure 10.4. **Agreement at a single automorphism orbit showing the effect of scaling on contribution to agreement.**

one makes sense. It means that the two graphs are identical. What does not make sense is achieving a value of zero, which we have shown does happen both in the individual orbit agreement level as well as in overall agreement. The interpretation of this is that these two graphs are as different as physically possible and we cannot do anything to make the graphs more different. We have discussed that the concept of "more different" is hard to perfectly define, especially for graphs. However, logic implies that we should be able to make these graphs infinitely more different, even if the difference in calculations is infinitesimally small.

In this chapter, we propose three reformulation of the GDD. We examine their performance in comparison to the original as well as asses their applicability more generally.

10.2 Methods

Three reformulations of the GDD algorithm were investigated. In all of the reformulated algorithms, the geometric mean was removed as a possible method to calculate the full graph agreement. Two features were varied between the algorithms: scaling step and algorithm structure.

10.2.1 Version 1

In Version 1, seen in Equation 10.35, the scaling step:

$$S_{\mathcal{G}}^{j}(k) = \frac{d_{\mathcal{G}}^{j}(k)}{k}, \tag{10.33}$$

was removed. This results in:

$$N_{\mathcal{G}}^{j}(k) = \frac{d_{\mathcal{G}}^{j}(k)}{T_{\mathcal{G}}^{j}}, \tag{10.34}$$

where $T_{\mathcal{G}}^{j} = \sum_{k=1}^{\infty} d_{\mathcal{G}}^{j}(k)$. The form of the distance equation remains the same as the original GDD (Equation 10.35). Removing the scaling step affects the resulting range of values, preventing a normalized outcome and making it impossible to turn the distance into an agreement.

$$D^{j}(\mathcal{G}_1, \mathcal{G}_2) = \frac{1}{\sqrt{2}} \left(\sum_{k=1}^{\infty} \left[N_{\mathcal{G}_1}^{j}(k) - N_{\mathcal{G}_2}^{j}(k) \right]^2 \right)^{1/2}. \tag{10.35}$$

10.2.2 Version 2

In the second method, Equation 10.37, scaling is not removed despite the noted problems, however the algorithm structure is edited. Thus $S_{\mathcal{G}}^{j}(k)$ is defined as in Equation 10.33. Then $N_{\mathcal{G}}^{j}(k)$ is defined such that:

$$N_{\mathcal{G}}^{j}(k) = \frac{S_{\mathcal{G}}^{j}(k)}{T_{\mathcal{G}}^{j}}, \tag{10.36}$$

which is the same as the definition of $N_{\mathcal{G}}^{j}(k)$ for the original GDD.

The new distance is structured similar to the distance used in both the RGF and the DDD. The form of the distance equation was changed to see if the many inconsistencies are a result of the overall idea or just an unfortunate side effect of the way the equation for

the agreement was designed. Thus the distance at any automorphism orbit j is defined as:

$$\mathcal{D}^j(\mathcal{G}_1, \mathcal{G}_2) = \sum_{k=k_1}^{k_2} |F_k(\mathcal{G}_1) - F_k(\mathcal{G}_2)| \tag{10.37}$$

$$k_1 = min\left(\delta(\mathcal{G}_1), \delta(\mathcal{G}_2)\right)$$

$$k_2 = max\left(\Delta(\mathcal{G}_1), \Delta(\mathcal{G}_2)\right)$$

where

$$F_k(\mathcal{G}) = \begin{cases} -log(N^j_{\mathcal{G}}(k)), & N^j_{\mathcal{G}}(k) \neq 0 \\ 0, & N^j_{\mathcal{G}}(k) = 0 \end{cases}. \tag{10.38}$$

In Equation 10.37, $\delta(\mathcal{G})$ is the minimum degree of graph $\mathcal{G}$ and $\Delta(\mathcal{G})$ is the maximum degree of that same graph. The equation can be rewritten as

$$\mathcal{D}^j(\mathcal{G}_1, \mathcal{G}_2) = \sum_{k=k_1}^{k_2} \left| \log\left(\frac{N^j_{\mathcal{G}_2}(k)}{N^j_{\mathcal{G}_1}(k)}\right)\right|. \tag{10.39}$$

10.2.3 Version 3

The final reformulated equation uses the same distance as in Version 2, but with the removal of scaling. Thus $N^j_{\mathcal{G}}(k)$ has the same definition as in Version 1. Overall, the distance is defined such that:

$$\mathcal{D}^j(\mathcal{G}_1, \mathcal{G}_2) = \sum_{k=k_1}^{k_2} \left| \log\left(\frac{N^j_{\mathcal{G}_2}(k)}{N^j_{\mathcal{G}_1}(k)}\right)\right|, \tag{10.40}$$

where:

$$N^j_{\mathcal{G}}(k) = \frac{d^j_{\mathcal{G}}(k)}{T^j_{\mathcal{G}}}. \tag{10.41}$$

10.2.4 Analysis of Performance

To test the accuracy of the reformulated methods 100 graphs were randomly selected from the 1000 total graphs for each of the nine model types. Similar to previous sampling procedures, 10 of these were designated as test graphs and the remaining 90 as comparison graphs. The median distance for each of the 10 graphs against all of the comparison graphs of a given type was calculated. The overall smallest median distance was determined to be the best fit. Only 100 graphs were used as opposed to the full 1000 seen in Section 7 with the goal of saving time, while still preserving statistical power and significance. Thus, the results are statistically sound (Burton *et al.* , 2006).

The reformulated algorithms were evaluated in the same ways as the original version in Section 7.1. Overall accuracy was calculated along with sensitivity, specificity, PPV, NPV, and both of the F-measures. Finally, the *S. cerevisiae* PPI network will be classified and the results interpreted using Bayes theorem.

10.3 Results

10.3.1 Model Graph Classification

All of the reformulated versions of the graphlet degree distribution work without exhibiting the idiosycrancies seen in the original version. Two of the versions correctly classified more than 70% of the model graph correctly. The third version classified only 33% of the model graphs correctly.

Version 1

The reformulated graphlet degree distribution version 1 removed the scaling step and kept the same algorithmic structure. This version had an accuracy of 72%. It correctly placed five model types into the correct category 100% of the time: GEO, LPA, RDG, RDS, and STI (Table 10.4). SMW was classified correctly 90% of the time. The other remaining models were classified correctly significantly less. AGV was only accurately classified 50%, while DMR was 10%. DMC was never classified correctly. Unlike other methods, the DMC and DMR model graphs were not spread among many different model types. Instead they were either DMC, RDG, or SMW. The DMR graphs were incorrectly classified as LPA,

RDG, or RDS.

Table 10.4. Classification accuracy of reformulated graphlet degree distribution version 1.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
Actual Class	AGV	50	-	-	10	40	-	-	-	-
	DMC	-	-	20	-	-	60	-	20	-
	DMR	-	-	10	-	10	40	40	-	-
	GEO	-	-	-	100	-	-	-	-	-
	LPA	-	-	-	-	100	-	-	-	-
	RDG	-	-	-	-	-	100	-	-	-
	RDS	-	-	-	-	-	-	100	-	-
	SMW	-	-	-	10	-	-	-	90	-
	STI	-	-	-	-	-		-	-	100

The values presented are the percent of model graphs classified into each category by GDD-V1. This version has the same structure as the original GDD, but with the scaling step removed.

If we look at the analysis of performance for Version 1, the F-macro is 0.6597 and the F-micro is 0.7263 (10.5). Only two models had low scores across the majority of the statistics: DMR and DMC (Figure 10.5). Both of these had low PPV and sensitivity, but high NPV and specificity. This is because very few, or none in the case of DMC, models of the given type were classified correctly. At the same time however, very few model graphs were incorrectly placed into these categories. The remaining graphs had high values across all of the statistics considered.

In Figure 10.5, we can see the groups mentioned in Section 7.3. The majority of the models are classified as Group 1, which are the models that were accurately classified often, but were also a very popular incorrect choice. Models in this include GEO, LPA, RDG, and RDS. DMC and DMR fall into Group 2, models that have low accuracy and are an unpopular incorrect choice. AGV is Group 3, low accuracy and never chosen incorrectly. Two models do not fall neatly into any category. These are STI and SMW. All of the STI graphs and nearly all of the SMW graphs were classified correctly and no graphs were incorrectly classified into either category.

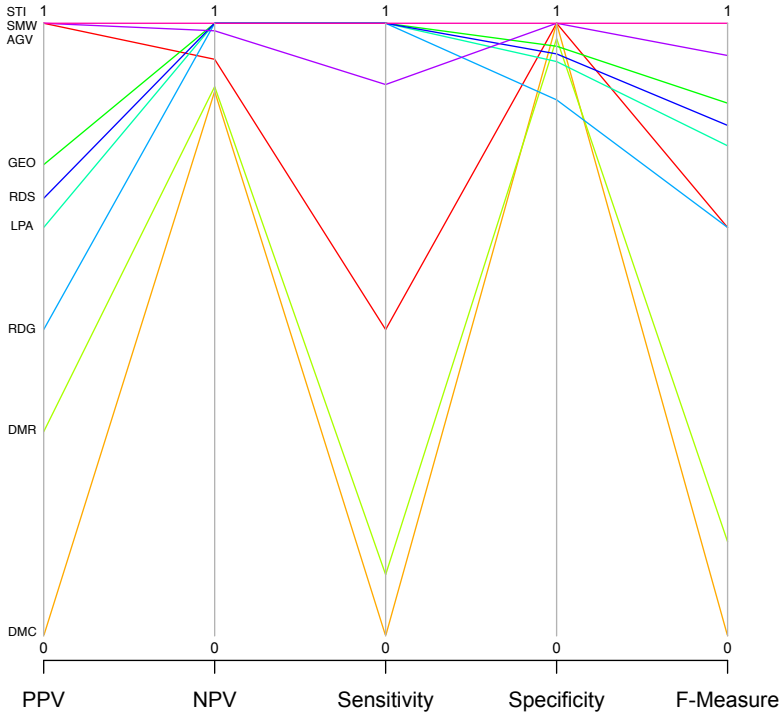


Figure 10.5. **Parallel coordinate representation of the graphlet degree distribution version 1 performance statistics.** This version has the same structure as the original GDD, but with the scaling step removed. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 10 models of each of the nine model graph types.

Table 10.5. Graphlet degree distribution version 1 analysis of performance.

Model	PPV	NPV	Sensitivity	Specificity	F-measure
AGV	1.0	0.9412	0.50	1.0	0.6667
DMC	0.0	0.8889	0.0	1.0	0.0
DMR	0.3333	0.8966	0.1	0.975	0.1538
GEO	0.7692	1.0	1.0	0.9625	0.8696
LPA	0.6667	1.0	1.0	0.9375	0.8
RDG	0.5	1.0	1.0	0.875	0.6667
RDS	0.7143	1.0	1.0	0.95	0.8333
SMW	1.0	0.9877	0.9	1.0	0.9474
STI	1.0	1.0	1.0	1.0	1.0
Average	0.6648	0.9683	0.7222	0.9667	0.6597
Global	0.7303	0.9999	0.7222	0.9667	0.7263

Results are calculated based on the classification of the 10 model graphs from each of the nine model types using the GDD-V1. This version has the same structure as the original GDD, but with the scaling step removed. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

Version 2

The reformulated graphlet degree distribution version 2 keeps the scaling step, but has the edited algorithm structure that resembles the RGF and DDD. This version had an accuracy of 33%. It correctly placed three model types into the correct category 100% of the time: GEO, RDS, and SMW (Table 10.6). No other model types were ever classified accurately, not even a single time. Interesting, the incorrect classification clustered into two model types: GEO and SMW. All of the LPA, RDG, STI graphs were classified as SMW while AGV, DMC, and DMR graphs were misclassified into both GEO and SMW.

Table 10.6. Classification accuracy of reformulated graphlet degree distribution version 2.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
Actual Class	AGV	-	-	-	10	-	-	-	90	-
	DMC	-	-	-	30	-	-	-	70	-
	DMR	-	-	-	10	-	-	-	90	-
	GEO	-	-	-	100	-	-	-	-	-
	LPA	-	-	-	-	-	-	-	100	-
	RDG	-	-	-	-	-	-	-	100	-
	RDS	-	-	-	-	-	-	100	-	-
	SMW	-	-	-	-	-	-	-	100	-
	STI	-	-	-	-	-		-	100	-

The values presented are the percent of model graphs classified into each category by GDD-V2. This version has the edited structure, but keeps the scaling step seen in the original GDD.

The way the incorrect models were classified into only two groups resulted in very low performance statistics. The F-macro for GDD-V2 is 0.2296 and the F-micro is 0.3333 (Table 10.7). The incorrect classifications also resulted in a very unique looking parallel coordinate representation where only four distinct lines are shown (Figure 10.6). AGV, DMC, DMR, LPA, RDG, and STI models all have the same performance statistics values, thus they are represented by the same line. These models all fall into Group 2. GEO falls into Group 1, which are models with high classification accuracy while still being a popular incorrect classification choice. SMW follows a similar pattern as GEO, but its values are much more extreme than those seen for GEO thus it does not adequately fit the pattern. The extreme fluctuations are because the majority of the misclassified graphs fall into this model type. RDS also does not fall neatly into any pattern.

Table 10.7. Graphlet degree distribution version 2 analysis of performance.

Model	**PPV**	**NPV**	**Sensitivity**	**Specificity**	**F-measure**
AGV	0.0	0.8889	0.0	1.0	0.0
DMC	0.0	0.8889	0.0	1.0	0.0
DMR	0.0	0.8889	0.0	1.0	0.0
GEO	0.6667	1.0	1.0	0.9375	0.8
LPA	0.0	0.8889	0.0	1.0	0.0
RDG	0.0	0.8889	0.0	1.0	0.0
RDS	1.0	1.0	1.0	1.0	1.0
SMW	0.1538	1.0	1.0	0.3125	0.2667
STI	0.0	0.8889	0.0	1.0	0.0
Average	0.2023	0.9259	0.3333	0.9167	0.2296
Global	0.3333	0.9999	0.3333	0.9167	0.3333

Results are calculated based on the classification of the 10 model graphs from each of the nine model types using the GDD-V2. This version has the edited structure, but keeps the scaling step seen in the original GDD. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

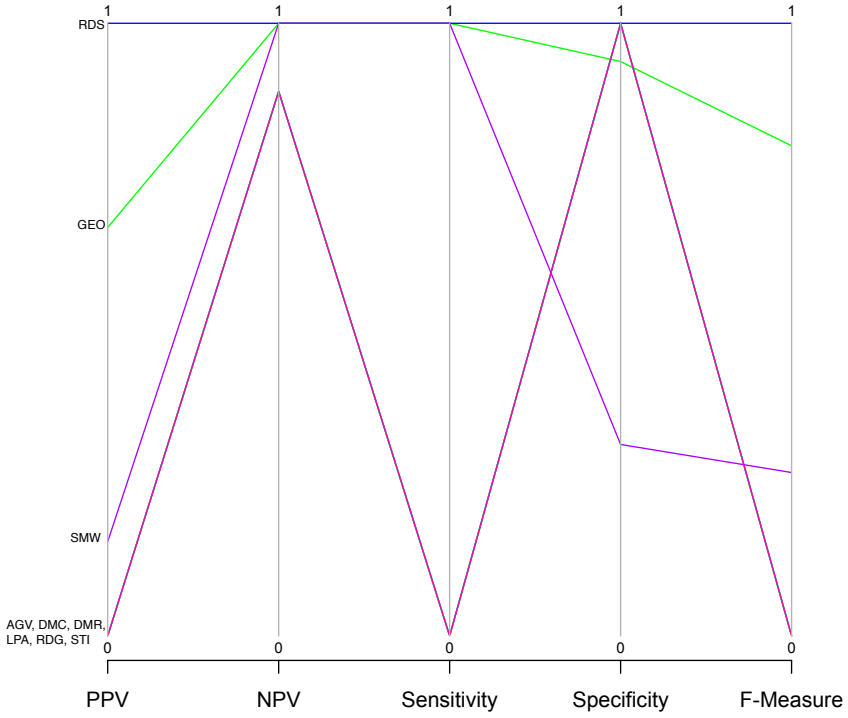


Figure 10.6. **Parallel coordinate representation of the graphlet degree distribution version 2 performance statistics.** This version has the edited structure, but keeps the scaling step seen in the original GDD. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 10 models of each of the nine model graph types.

Version 3

The reformulated graphlet degree version 3 removed the scaling step and also utilized the edited algorithm structure. It has an accuracy of 76%, the highest of all the reformulated versions considered (Table 10.8. Only four models were correctly classified 100% of the time, however, as opposed to Version 1's five correctly classified methods. These model types are GEO, RDG, RDS, and STI. Three models were classified correctly at or above 80% of the time: AGV, LPA, and SMW. Only two model types were classified extremely poorly. These are DMC and DMR. None of the DMC models were classified correctly and only 20% of the DMR models were.

Table 10.8. Classification accuracy of reformulated graphlet degree distribution version 3.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
Actual Class	AGV	90	-	-	10	-	-	-	-	-
	DMC	-	-	-	70	-	-	30	-	-
	DMR	-	-	20	-	-	10	50	-	20
	GEO	-	-	-	100	-	-	-	-	-
	LPA	10	-	-	-	90	-	-	-	-
	RDG	-	-	-	-	-	100	-	-	-
	RDS	-	-	-	-	-	-	100	-	-
	SMW	-	-	-	20	-	-	-	80	-
	STI	-	-	-	-	-		-	-	100

The values presented are the percent of model graphs classified into each category by GDD-V3. This version has the edited algorithm structure and the scaling step was removed.

Version 3 has an F-macro of 0.7013 and F-micro of 0.7556 (Table 10.9). Several of the model types have high values for every statistical performance measure. These are AGV, LPA, RDG, SMW, and STI. GEO and RDS have slightly lower PPV than the afore mentioned models despite having similar statistics for the remaining measures. DMC and DMR both perform poorly due to extremely inaccurate model classification.

Figure 10.7 shows the parallel coordinate representation of the performance statistics for Version 3. GEO, RDG, RDS, and STI fall into Group 1. DMC is alone in Group 2 while DMR is alone in Group 3 and SMW in Group 4. Two models, AGV and LPA, do not fall into any categories.

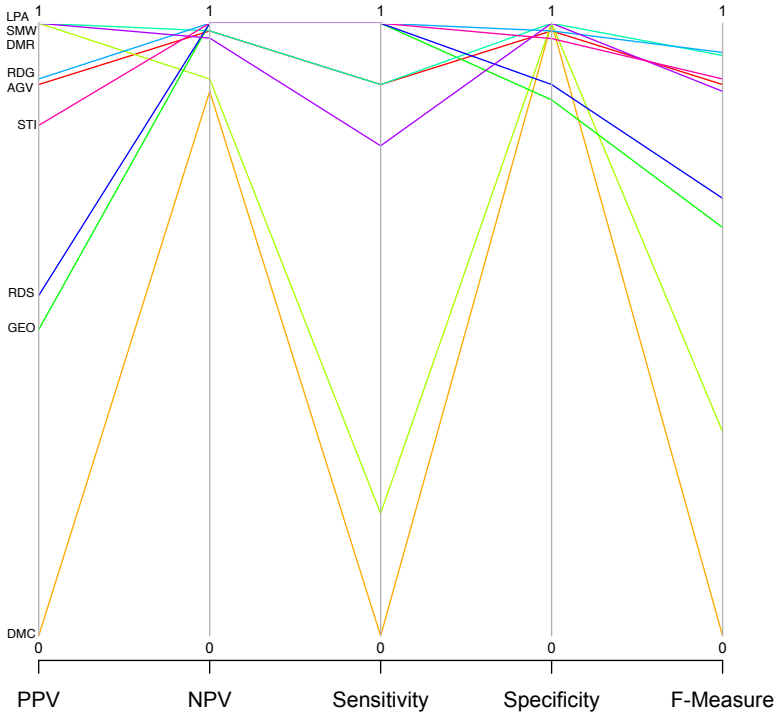


Figure 10.7. **Parallel coordinate representation of the graphlet degree distribution version 3 performance statistics.** This version has the edited algorithm structure and the scaling step was removed. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated using the classification results of the 10 models of each of the nine model graph types.

Table 10.9. Graphlet degree distribution version 3 analysis of performance

Model	PPV	NPV	Sensitivity	Specificity	F-measure
AGV	0.9	0.9875	0.9	0.9875	0.9
DMC	0.0	0.8889	0.0	1.0	0.0
DMR	1.0	0.9091	0.2	1.0	0.3333
GEO	0.5	1.0	1.0	0.875	0.6667
LPA	1.0	0.9877	0.9	1.0	0.9474
RDG	0.9091	1.0	1.0	0.9875	0.9524
RDS	0.5556	1.0	1.0	0.9	0.7143
SMW	1.0	0.9759	0.8	1.0	0.8889
STI	0.8333	1.0	1.0	0.975	0.9091
Average	0.7442	0.9721	0.7556	0.9694	0.7013
Global	0.7556	0.9999	0.7556	0.9694	0.7556

Results are calculated based on the classification of the 10 model graphs from each of the nine model types using the GDD-V3. This version has the edited algorithm structure and the scaling step was removed. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

10.3.2 Comparison of the Original Graphlet Degree Distribution to the Reformulated Versions

Two of the reformulated versions of the graphlet degree distribution performed extremely well, Table 10.10. In fact, both versions 1 and 3 performed better than the original version (72%, 76% v 68%). The majority of the improvement came in the classification of AGV and SMW. Looking at AGV, the original GDD only classified 40% correctly. Version 1 classified 50% and version 3 classified 90% correctly. For SMW, the original GDD classified only 60% of the graphs correctly. Version 1 classified 90% and version 3 classified 80% correctly. Unfortunately, none of the reformulated versions classified any of the DMC graphs correctly, which is a common theme among all of the algorithms considered. Version 3 classified 10% more DMR graphs correctly than both the original and version 1, each of which classified 10% correctly.

Version 2, on the other hand, had an overall classification accuracy of only 33%. This is the lowest of any method considered previously and simply from this statistic Version 2 can be dismissed as a viable replacement option for the original GDD.

Focusing on the performance statistics for the classification of each individual model shows that using different classifiers results in different group designations for many of the model graphs (Table 10.11). GEO and DMC are the only model types that remain in the same group for all versions, Group 1 and Group 2 respectively. In terms of groupings,

Table 10.10. Comparison of the classification accuracy of the reformulated versions of the graphlet degree distribution.

	Classification Accuracy			
Model	Original	V1	V2	V3
AGV	40	50	0	90
DMC	0	0	0	0
DMR	10	10	0	20
GEO	100	100	100	100
LPA	100	100	0	90
RDG	100	100	0	100
RDS	100	100	100	100
SMW	60	90	100	80
STI	100	100	0	100
Average	68	72	33	76

The values in the table indicate the percentage of the given model graph that was accurately classified by the classification method. Four classification methods are shown. Original refers to the original graphlet degree distribution using arithmetic mean. The remaining three classifiers are the three reformulated versions of the original (e.g. V1 refers to version 1).

version 2 appears to be an anomaly. There are many more similarities in group placement between the original GDD and versions 1 and 3 than between version 2. If we consider only version 1 and 3, we can add LPA, RDG, and RDS to the list of models that do not change groups. Since Group 1 is models with high accuracy this indicates that these models were all classified with high accuracy no matter the classifier used.

Table 10.11. Model graph groupings based on performance statistics.

	Group 1	Group 2	Group 3	Group 4	No Group
Original	GEO, LPA, RDG, RDS, STI	DMC	AGV, DMR, SMW	-	-
Version 1	GEO, LPA, RDG, RDS	DMC, DMR	AGV	-	SMW, STI
Version 2	GEO	AGV, DMC, DMR, LPA, RDG, STI	-	-	RDS, SMW
Version 3	GEO, RDG, RDS, STI	DMC	DMR	SMW	AGV, LPA

The groups listed correspond to those presented in Figure 7.13. Results are based on the patterns of performance statistics presented in Figures 10.5, 10.6, and 10.7. Original refers to the original graphlet degree distribution using arithmetic mean. The remaining three classifiers are the three reformulated versions of the original GDD.

Finally, if we examine the analyses of performance for the original GDD as well as version 1 and 3, we see that version 3 has the highest F-macro, 0.7013 compared to 0.6597 for version 1 and 0.5444 for the original (Figure 10.8). It also has the highest F-micro (0.7556 v 0.7263, 0.6778). In fact, version 3 has higher global statistics than any of the other versions of GDD including the original. Due to its better classification accuracy, higher

performance statistics, and lack of inconsistent results, we can conclude that reformulated graphlet degree distribution reformulation version 3 is the best replacement for the original GDD.

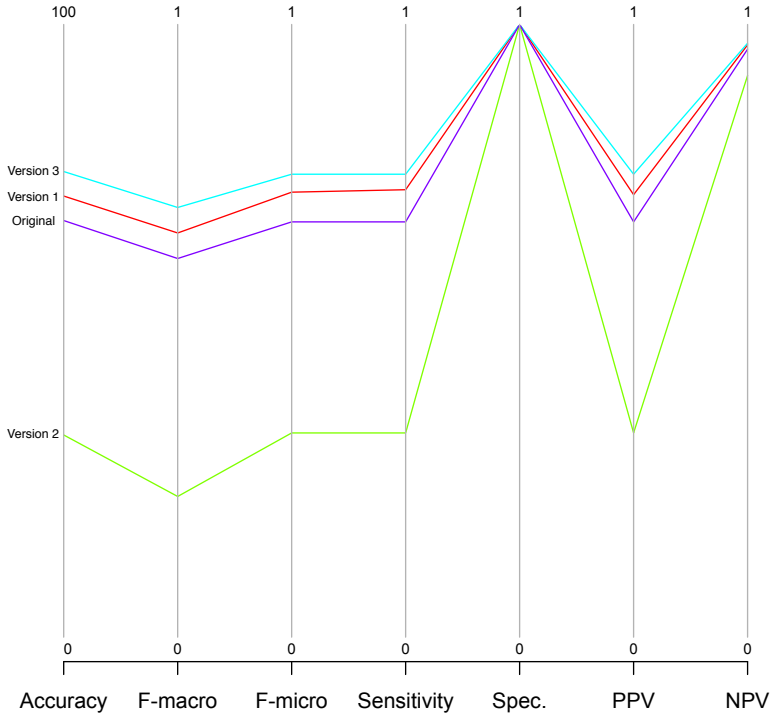


Figure 10.8. **Parallel coordinate comparison of original and reformulated versions of the graphlet degree distribution performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and both F-measures. Values presented for sensitivity, specificity, PPV and NPV are the global results of the classification results of the 100 models of each of the nine model graph types.

10.3.3 Saccharomyces cerevisiae PPI Network Classification

Under the original GDD using the arithmetic mean, the model type declared the best fit for the *S. cerevisiae* PPI network was the GEO model graph. There was a 100% chance that if the *S. cerevisiae* PPI network really is GEO then it would be classified as such and a 20% chance that the *S. cerevisiae* PPI network would be classified as GEO when it really is a different model. With such a high probability of incorrect classification, these results are difficult to trust.

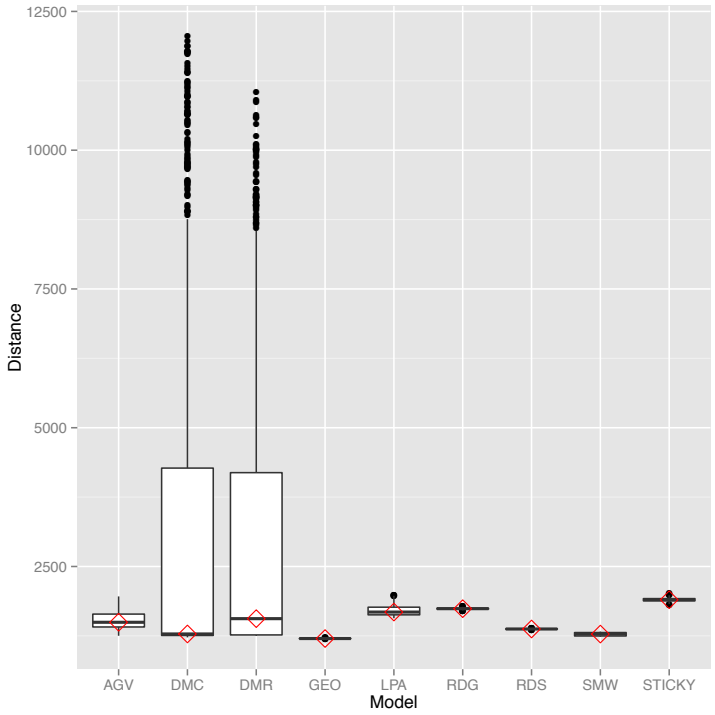


Figure 10.9. ***S. cerevisiae* PPI network Classification by reformulated graphlet degree distribution version 3.** The figure shows the results of comparing the empirical *S. cerevisiae* PPI network against the 1000 model graphs of each of the nine types. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances.

The best fit for the empirical network under Version 3 of the reformulated GDD is also GEO, though the results are close (Figure 10.9). Using Bayes theorem, we conclude that there is a 100% chance that if the *S. cerevisiae* PPI network really is GEO, then it will be classified as such. There is a 12.5% chance that even if the *S. cerevisiae* PPI network is not really GEO, it will still be classified as GEO. This is lower than the chance of incorrect

classification given by the original GDD.

$$\begin{aligned}\Pr(\text{classified as GEO} \mid \text{GEO}) &= \frac{\Pr(\text{GEO} \mid \text{classified as GEO}) \cdot \Pr(\text{classified as GEO})}{\Pr(\text{GEO})} \\ &= \frac{\frac{100}{200} \cdot \frac{200}{900}}{\frac{1}{9}} \\ &= 1 \end{aligned} \tag{10.42}$$

$$\begin{aligned}\Pr(\text{classified as RDG} \mid \text{not RDG}) &= \frac{\Pr(\text{not RDG} \mid \text{classified as RDG}) \cdot \Pr(\text{classified as RDG})}{\Pr(\text{not RDG})} \\ &= \frac{\frac{100}{200} \cdot \frac{200}{900}}{\frac{8}{9}} \\ &= 0.125 \end{aligned} \tag{10.43}$$

Table 10.12 shows the rankings of the nine model graph types when classified by the original GDD using arithmetic mean and when classified using version 3 of the reformulated GDD. The lists have a Kendall's W of 0.425 indicating that they match less than 50% of the time. In fact, only one of the model graphs appear at the same ranking in both lists: GEO. Two model types appear only one position off: AGV and RDS. Overall, it appears that the reformulated GDD, while using the same basic premise as the original GDD, does not share much in common in terms of classification.

Table 10.12. Ordered rankings of the model graphs based on fit for S. cerevisiae PPI network using the original graphlet degree distribution and reformulated version 3.

	Original	Version 3
1	GEO	GEO
2	**RDG**	**SMW**
3	**RDS**	**DMC**
4	**AGV**	**RDS**
5	**LPA**	**AGV**
6	**DMC**	**DMR**
7	**STI**	**LPA**
8	**SMW**	**RDG**
9	**DMR**	**STI**

Rankings of model graphs are determined based on the median distance of the S. cerevisiae PPI network to each model graphs of the given type. The median smallest distance is ranked first. Models in bold show up in different places across the two lists. Items not in bold do not change position when the reformulated graphlet degree distribution version 3 is used in place of the original.

10.4 Conclusions

We have shown that there are too many inconsistencies in the graphlet degree distribution for it to be considered as a classifier for the *S. cerevisiae* PPI network. We have addressed these inconsistencies and proposed three reformulated versions of the algorithm. All of these reformulations keep the same basic premise in mind (i.e. comparing graphs based on graphlet degree distribution), but they also provide more consistent results. Of the three versions considered, one performed so poorly, version 2, that it was immediately dismissed as a possible replacement for the original GDD. Version 2 produced substantially different results than were seen in the other two reformulations. One interesting aspect of the results is that graphs were only classified into one of three categories: GEO, RDS, and SMW. Incorrectly classified model graphs were only classified as GEO or SMW with SMW taking approximately 92% of the misclassified graphs. We can only speculate on why version 2 classified graphs so differently than the other versions but we can conclude that clearly some combination of scaling combined with the structure of the algorithm limited its ability to discern differences between the model graphs.

The two other reformulations, version 1 and version 3, produced very similar values for all of the statistics examined. They both reported higher accuracies than the original GDD (72%, 76% v 68%), as well as higher F-measures. Since version 3 has a slightly higher value for every statistic considered than version 1, it is deemed the best replacement for the original GDD. This shows that the method is greatly improved by removing the scaling step and by editing the structure of the algorithm. The first step is of particular importance because it will make the graphlet degree distribution more generalizable. The scaling step was originally created because it was found that graphlets of high degree were often noise (Przulj, 2007). While this may be true for PPI networks, it is not necessarily true of networks of other types. By removing this step, we are forcing biologists to clean up their networks (i.e. eliminating noise) before classifying which should lead to better results. We are also allowing the method to be applied to other network classification problems.

Chapter 11

Designing the Cross Scoring Algorithm

One of the main issues with the previously examined network classification algorithms is that they only look at features of one scale. Characteristic curve and degree distribution distance both look at large-scale features while relative graphlet frequency and degree distribution distance look at small-scale. None of these methods use a diverse set of features to classify the networks nor do they have an easily interpretable structure. Therefore, we propose a new method that is based on the idea that the best fitting model graph is the one that averages the highest ranking across a set of criteria, as opposed to highest ranking on only one criteria. This new classification algorithm is referred to as Cross Scoring (CS).

The overall idea for the CS algorithm is simple. First, a set of graphical features are selected. Average values for these features are calculated across the nine types of model graphs. These values are then ranked by their distance from the measured value of the test graph under examination. The test network is either the real world network in which one is interested in finding a match, or one of the numerous model graphs that were deemed test networks. From there, the model type with the closest value is awarded one point, second closest receives two points, and so on. These steps are performed for each network measure under consideration. When all of the measures have been ranked, the points are added up. The model graph category with the lowest score is deemed the best fit.

In this chapter, we propose several ways to design the algorithm, discuss the pros and cons of each, and then declare a final algorithm. In Chapter 12, we further refine aspects of cross scoring and explain these aspects through a simple example. Finally in Chapter 13, we will show how the algorithm can be applied to the real-world classification problem that has previously been discussed: classification of the *S. cerevisiae* PPI network.

11.1 Methods

Even though the idea behind the Cross Scoring classifier is straightforward, there are many potential choices to be made when designing the algorithm. The first choice is to decide which measure of center to use. This can be either mean or median. Another choice is

whether to round values that are very close to the desired empirical value to that value, thus resulting in a distance of zero. In other words, should models that produce measure values within 1% or 5% of the empirical value be rewarded for their near accuracy? Other factors to consider are whether scoring should be linear and whether measures of spread, either standard deviation or IQR depending on the measure of center, should be applied. Further detail for these factors can be seen in the following subsections.

11.1.1 Measures of Center and Spread

Two different measures of center (mean and median) are tested to see which achieves an optimal result. In addition to the required measure of center, a measure of spread can be added. This is either standard deviation if the measure of center is mean or interquartile region (IQR) if the measure of center is median. These can be used to break ties in a model. For instance, if Model A and Model B both have the same value for a measure, then the two model types will receive an equal score, which is not necessarily warranted. If we consider the measure of spread, a smaller measure indicates that the growth mechanism is more likely to reproduce the desired value, or at least one close to it, than the model with a larger measure of spread. Thus if Model A has a smaller measure of spread, it will be ranked above Model B.

11.1.2 Nonlinear Scoring

In the basic design of CS, one point is earned if the model type has the closest measure value to the empirical value, two for second closest, and so on. Scoring can be modified in order to add additional penalties to models that present with values dramatically different than the desired value.

Two different nonlinear scoring schemes were evaluated. For each of the nonlinear schemes, we broke the scoring into groups (Table 11.1). In both schemes, models in first and second place are scored normally, given one and two point respectively. Those placing between third and fifth place are scored with an additional two points. In the first scheme, models placing sixth and higher are given an additional four points. In the second scheme, models in fifth through eight place receive an additional five points and the model type

Table 11.1. Generalized nonlinear scoring schemes 1 and 2 for cross scoring algorithm depicted by rank ranges.

	Score	
Rank i	Scheme 1	Scheme 2
$i < \frac{n}{3}$	i	i
$\frac{n}{3} \leq i < \frac{2n}{3}$	$i+2$	$i+2$
$\frac{2n}{3} \leq i < n$	$i+4$	$i+5$
$n \leq i$	$i+4$	$i+9$

This table displays the number of points prescribed to different ranges of rankings for two different nonlinear scoring systems. Rank i is the location that the model graph is ranked in comparison to other model graphs based on some graph measure. Score is the number of points awarded to the graph of rank i. The variable n is the number of model graphs in the comparison. This table is generalized to work for any number of model graphs.

in ninth place receives an additional nine points. Exact scores for each place in the two schemes can be seen in Table 11.2. The exact details regarding the divisions and the number of additional points added is arbitrary. This was done to see if it provided any increase in accuracy. If an increase in classification accuracy is seen, more precise schemes can be developed.

Table 11.2. Nonlinear scoring schemes 1 and 2 for cross scoring algorithm.

	Score	
Rank	Scheme 1	Scheme 2
1	1	1
2	2	2
3	5	5
4	6	6
5	7	7
6	10	11
7	11	12
8	12	13
9	13	18

This table displays the number of points prescribed to each rank based on a total of nine model graphs. Rank is the location that the model graph is ranked in comparison to other model graphs based on some graph measure. Score is the number of points awarded to the graph. Scheme 1 and scheme 2 are the two unique scoring systems presented in Table 11.1.

11.1.3 Zeroing

The scoring system can be modified in more subtle ways than moving to nonlinear scoring. If we use zeroing in the model, than any test graph that achieves the exact same value as the empirical value is given no points, instead of the standard one point for being in first

place. If the test graph is in first place, but the distance is not zero, then one point is still given. This is done to further reward models that exactly reproduce desired values. The graph with the second closest value is still awarded two points.

11.1.4 Approximations

Many of the features were particularly difficult for the model graphs to mimic, especially centrality and other connectivity measures. One way to further reward model graphs that were able to get close to the empirical value when the majority of other graphs were far is to allow values within 1% or 5% to be considered mathematically equal to the empirical value. This artificially increases the number exact matches to the empirical value. When combined with zeroing, the models that are close get further rewarded.

11.1.5 Tie Breaking

No matter how the algorithm is designed, ties for overall best fitting model are inevitable. Ties are broken by counting the number of low rankings, those that give a high score, that the model receives. Example 5 provides a concrete explanation of how this works.

Example 5 Consider three model graph types (A, B, and C) along with any five measures such that the models receive the rankings:

- A: $(1, 1, 2, 3, 1)$
- B: $(2, 2, 1, 1, 2)$
- C: $(3, 3, 3, 2, 3)$

Model A and Model B both have a score of 8, while Model C has a score of 14. Since the lowest score is considered the best fit for the test network, Model C can be eliminated as a possible option. Table 11.3 shows the number of times each model is classified at each of the three rankings. Model A has one third place ranking and Model B has zero. Therefore, Model B is considered the best fit.

Table 11.3. Counts of rankings for Model A and Model B in Example 5.

	Count	
Rank	**Model A**	**Model B**
1	3	2
2	1	3
3	1	0

This table shows the number of times Model A and Model B are ranked first, second, or third place in Example 5. Count signifies the number of times each model is ranked a certain way.

The reason that the number of low rankings is considered detrimental, as opposed to the number of high rankings being positive, is that we chose to reward models that perform consistently across all features. We also chose not to consider a model with three first best finishes significantly better than a model with only two. Looking at the lower rankings gives a better idea of a model type's performance on the features considered important.

11.2 Data

Two forms of data are needed to perform cross scoring. First, we need graph data to classify and graph data to compare. For this, 1000 graphs of each of the nine model types were used. From each set of 1000 model graphs, 100 were designated as test graphs that need to be classified and the remaining 900 as comparison graphs.

The second type of data needed is the list of features to use. Two different lists of features were used in order to get an idea of how each of the potential algorithms worked in different scenarios (Table 11.4). The first measure list is based on the Nicosia criteria (Nicosia *et al.* , 2013). Nicosia argues that nodal properties such as degree, average neighbor degree, and clustering coefficient are extremely important in revealing the existence of local and global graph features and thus can be used to distinguish between different categories of graphs.

The second measure list is based on biologically significant features. In Section 4.2 the biologically significant features were listed as density, transitivity, average degree, and assortativity. To that list we add average neighbor degree and average shortest path length to finish off the biologically significant sequence. We use two lists to confirm that we are designing the best overall implementation of cross scoring and not designing it to work with only one list of features.

Table 11.4. Initial measures used to determine the best cross scoring algorithm implementation.

Nicosia (Nic)	**Biologically Significant** (Bio)
Average Degree	Average Degree
Average Clustering Coefficient	Density
Average Neighbor Degree	Average Neighbor Degree
	Transitivity
	Assortativity
	Average Shortest Path Length

Cross scoring requires a list of measures in order to classify networks. In order to find the best classifier design, two measure lists were used. Nicosia is based on the Nicosia criteria (Nicosia *et al.* , 2013) and biologically significant are measures that have simple biological interpretations.

11.3 Results

We analyzed 48 different algorithm implementations. This is not a full representation of all possible combination of algorithmic features discussed in Section 11.1, because some algorithms performed so poorly that it was not necessary to continue down their path. These formulations were tested on two sequences of features, the Nicosia criteria, *Nic*, and the biologically significant measures, *Bio* (Table 11.4).

Tables 11.6 and 11.7 show the classification results of various combinations of factors that comprise the CS algorithm. The first factor in each table is measure of center, i.e. mean or median. This factor is used as the basis to separate the results into two unique tables. The next is whether a measure of spread is used. In the median table this measure of spread is the interquartile range, and in the mean table it is the standard deviation. The *Scoring* column represents the type of scoring used. This can be either linear or nonlinear. If it is nonlinear it falls into either scheme 1 or scheme 2, as discussed in the methods section (Tables 11.1 and 11.2). The *approximation* (*Approx.*) column indicates whether values within 1% or 5% are considered equal to the empirical value. *Zeroed* corresponds to whether values equal to the empirical value are rewarded by not receiving any points at all.

The final three columns deal with the classification results. The *Measure* columns indicates which measure list was used and the accuracy of the model using that list is reported in the next column. Finally, the *Average Accuracy* over the two measure lists considered is computed in the *Average* column.

11.3.1 Mean Results

Table 11.5 shows the results for the twelve algorithms created when the measure of center used is the arithmetic mean. The first thing to note is that many of the features used do not have any affect on the results of the model. Nothing changes when measure of spread is used or when zeroing is applied. This allows the full table to be pared down to the more manageable Table 11.6. Here is it obvious that when the mean is used, linear scoring results in the best average accuracy. Approximately 69.5% of the 900 graphs classified were done so correctly. When nonlinear scoring was applied, both schemes performed with nearly the same accuracy (53.5% v 53%).

Table 11.5. Results of model graph classification based on variations of the cross scoring algorithm using mean as the measure of center.

	Measure of Center	**Measure of Spread**	**Scoring**	**Zeroed**	**Measures**	**Accuracy**	**Average Accuracy**
1	Mean	No	Linear	No	Nic	0.72	0.695
2					Bio	0.67	
3				Yes	Nic	0.72	0.695
4					Bio	0.67	
5			Scheme 1	No	Nic	0.38	0.535
6					Bio	0.69	
7				Yes	Nic	0.38	0.535
8					Bio	0.69	
9			Scheme 2	No	Nic	0.38	0.53
10					Bio	0.68	
11				Yes	Nic	0.38	0.53
12					Bio	0.68	
13		Yes	Linear	No	Nic	0.72	0.695
14					Bio	0.67	
15				Yes	Nic	0.72	0.695
16					Bio	0.67	
17			Scheme 1	No	Nic	0.38	0.535
18					Bio	0.69	
19				Yes	Nic	0.38	0.535
20					Bio	0.69	
21			Scheme 2	No	Nic	0.38	0.53
22					Bio	0.68	
23				Yes	Nic	0.38	0.53
24					Bio	0.68	

Table displays the average accuracies for several cross scoring variants. All of the variations described here use mean as their measure of center. Median results are described in Table 11.7. The first four columns, not counting the line number column, describe features that may or may not be present in the algorithm. The fifth column, Measures, indicates which measure list the algorithm is applied to and accuracy is the classification accuracy. Average accuracy is the average of the accuracy values across the two measure lists.

Graphs classified using the Nicosia measure list were classified more accurately when the scoring was done linearly. This is a difference in accuracy of approximately 5% (72% v 67%). Those model graphs classified under the biologically significant measure list were more accurate with nonlinear scoring. Under both scheme 1 and scheme 2, Nicosia classification only worked correctly 38% of the time while biologically significant classification worked correctly 69% and 68%, respectively. The overall accuracy for when the mean is used is 58.7%.

Table 11.6. Summarized accuracies of model graph classification based on variations of the cross scoring algorithm using mean as the measure of center.

Measure of Center	**Average Accuracy**	**Scoring**	**Average Accuracy**
Mean	0.587	Linear	0.695
		Scheme 1	0.535
		Scheme 2	0.53

Table display the average accuracies for several cross scoring variants. All of the variations described here use mean as their measure of center. Median results are described in Table 11.7. There are two features described, measure of center and scoring. Each feature column is followed by the average accuracy of all of the models that have the listed feature. The average accuracies come from the classification accuracies in Table 11.5.

11.3.2 Median Results

When the measure of center utilized was the median, variations of the algorithm had more impact on classification accuracy. While zeroing continued to have no impact, changing scoring, use of approximation, and adding in a measure of spread all had an affect on the results (Table 11.7).

Looking first at the different scoring schemes used without adding in IQR, linear scoring is the most accurate for every level of approximation. Within linear scoring, not using any approximation resulted in the highest accuracy, 78.5%. When 1% approximation was applied this values decreased to 77% and further decreased to 70% with 5% approximation. For both nonlinear scoring schemes, 5% approximation was more accurate than the other choices.

When measure of spread is applied, linear scoring with no approximation is the most accurate combination of features with 78.5% of graphs classified correctly. This is the same accuracy seen for this feature combination as when IQR was not included. Though the results for linear scoring are very similar, even for the approximated models, the results are not quite as similar for nonlinear scoring models. Previously nonlinear scoring with 5% approximation was the most accurate. When the measure of spread is added in, this changes to 1% approximation.

Table 11.8 shows overall averages of each grouping of algorithms. Linear scoring with or without IQR has a higher average accuracy, correctly classifying 75.2% or 76.2% of graphs when measure of spread is not used or is used, respectively. Averaging all of the scoring schemes together, models without measure of spread have an average accuracy of 68.7% while using measure of spread has an average accuracy of 68.1%. The overall accuracy of using the median is 68.4%.

Table 11.7. Results of model graph classification based on variations of the cross scoring algorithm using median as the measure of center.

	Measure of Center	Measure of Spread	Scoring	Approx.	Features	Accuracy	Average
1	Median	No	Linear	NA	Nic	0.78	0.785
2					Bio	0.67	
3				1%	Nic	0.76	0.77
4					Bio	0.78	
5				5%	Nic	0.68	0.70
6					Bio	0.72	
13			Scheme 1	NA	Nic	0.51	0.595
14					Bio	0.68	
15				1%	Nic	0.60	0.625
16					Bio	0.65	
17				5%	Nic	0.72	0.75
18					Bio	0.78	
25			Scheme 2	NA	Nic	0.51	0.575
26					Bio	0.64	
27				1%	Nic	0.60	0.625
28					Bio	0.65	
29				5%	Nic	0.73	0.755
30					Bio	0.78	
37		Yes	Linear	NA	Nic	0.78	0.785
38					Bio	0.79	
39				1%	Nic	0.77	0.78
40					Bio	0.79	
41				5%	Nic	0.67	0.72
42					Bio	0.77	
49			Scheme 1	NA	Nic	0.51	0.595
50					Bio	0.68	
51				1%	Nic	0.64	0.69
52					Bio	0.64	
53				5%	Nic	0.61	0.66
54					Bio	0.71	
61			Scheme 2	NA	Nic	0.51	0.575
62					Bio	0.64	
63				1%	Nic	0.64	0.675
64					Bio	0.71	
65				5%	Nic	0.61	0.645
66					Bio	0.68	

Table displays the average accuracies for several cross scoring variants. All of the variations described here use mean as their measure of center. Median results are described in Table 11.5. The first four columns, not counting the line number column, describe features that may or may not be present in the algorithm. The fifth column, Measures, indicates which measure list the algorithm is applied to and accuracy is the classification accuracy. Average accuracy is the average of the accuracy values across the two measure lists.

11.3.3 Comparison of Mean and Median Results

If we calculate the average value for the median models without including 1% and 5% approximations in order to compare them to the mean models, the average accuracy is 65.2%, which is 6.5% higher than the average accuracy across all of the mean models. Thus the median outperforms the mean. Therefore, approximations were not considered for the mean models because it was clear from the beginning that the median models were outperforming them. Therefore, it is straightforward to choose median as the measure of center to use for the final algorithm implementation. It is important to note, that if the values of the measures under consideration are roughly normally distributed, then the mean and the median should be approximately the same.

Table 11.8. Summarized accuracies of model graph classification based on variations of the cross scoring algorithm using median as the measure of center.

	Measure of Center	**Avg. Accuracy**	**Measure of Spread**	**Avg. Accuracy**	**Scoring**	**Avg. Accuracy**
1	Median	0.684	No	0.687	Linear	75.2
13					Scheme 1	0.657
25					Scheme 2	0.652
37			Yes	0.681	Linear	0.762
49					Scheme 1	0.648
61					Scheme 2	0.632

Table display the average accuracies for several cross scoring variants. All of the variations described here use mean as their measure of center. Median results are described in Table 11.5. There are three features described, measure of center, measure of spread, and scoring. Each feature column is followed by the average accuracy of all of the models that have the listed feature. The average accuracies come from the classification accuracies in Table 11.7.

The choices of which other algorithm features to include are less straightforward than the choice for measure of center. The difference between measure of spread is a sixth of a percent in favor of no IQR (68.7% v 68.1%), however the highest average accuracy using linear scoring is one percent in favor of including IQR. Since linear scoring always performs better on average than either approximation, we choose linear scoring. With linear scoring, average accuracy is always higher without approximations. Thus, the highest accuracy was achieved with linear scoring and no approximations made with or without IQR. In order to include as many features into the algorithm, and because including IQR does not appear detrimental with linear scoring, we choose the model shown in lines 37/38 of Table 11.7

as the final algorithm for cross scoring. This algorithm makes use of median, IQR, linear scoring and makes no approximations.

11.4 Discussion

In this chapter, we proposed a novel network classification algorithm based on the idea that the model type which performs consistently across several graph measures should be considered a better match than one that performs very well on some features and very poorly on others. We then proceeded to discuss the design of this new classifier, called the cross scoring algorithm. When designing the algorithm, several features were considered for addition into the method, including: measure of center, measure of spread, the structure of the scoring, zeroing, and approximations.

When the mean was used, there was no difference between using the standard deviation to differentiate between graphs with the same measure values. We speculate that this is because using the mean on data that are not normally distributed results in skewed values. Model graphs that produced skewed results also tended to have larger standard deviations. Note that this fact is particular to the situation at hand due to the way in which the graph growth mechanisms are structured. Some growth mechanisms resulted in graphs with very similar values, while others resulted in graphs that differed greatly in terms of structure and thus in terms of measure values. This

The median is robust to skewed data, which is why medians are often used to describe non-normal distributions. Therefore, the IQR adds additional information while the standard deviation used in collaboration with the mean most likely results in redundant information, thus resulting in a lack of effect for the mean models, but a positive effect on the median models.

When median scoring was used, the 5% approximations outscored the 1% and no approximation, when no measure of spread and nonlinear scoring were used. We speculate that this is because nonlinear scoring punishes model graphs that produce values that are different than the empirical measure value. At the same time, 5% approximations reward model graphs that create values very close to the empirical value. When these two features

are used together, they essentially amplify each other, resulting in the highest classification accuracy for nonlinear scoring without IQR.

When nonlinear scoring is used in conjunction with IQR, the 1% approximations become more accurate than the 5%, which is likely due to the increased information provided by the IQR. This increased information eliminates ties. Most likely, the use of the 5% approximations without the ability for graphs to tie results in graphs being falsely ranked higher than they should be due to the increased range of graphs being declared exact matches to the empirical value.

It is essential to note that there are limitations with the design of the algorithm. In particular, it was only applied to one type of problem. Even though the model graphs produce a variety of measure distributions, a different scoring scheme may provide better results for a different problem. With that stated, we proceed with this algorithm because it appears to be the best fit for the classification problem at hand and does not seem fitted to the measure list used. In Chapter 12, we discuss how the measure list is calculated and then in Chapter 13 we test the cross scoring metric's ability to accurately classify the model graphs, before applying it to the *S. cerevisiae* PPI network.

Chapter 12

Determining the Cross Scoring Measure List

Once the cross scoring algorithm has been designed, there is another piece that must be determined before it can be directly applied. This is the determination of the measure list. The determination of this list takes place in the first of cross scoring's two stages: the build stage. The second stage, test stage, is discussed in greater detail in Chapter 13 where the algorithm is applied to the *S. cerevisiae* PPI network.

In this chapter, we explain the build stage and present three possible methods for determining the measure list: macro-scoring, micro-scoring, and importance scoring. Each method results in different answers to the classification problem, but all have high levels of accuracy. The methods are described along with an example that demonstrates how each works.

12.1 Methods

After determining the structure of the algorithm (Chapter 11), the next step in the process of building the full cross scoring model is to determine the best set of graph measures or features to be used in the classification step. We are looking to determine the selection of graph measures, or features, whose presence leads to the highest level of accuracy. Ideally, a simpler model with fewer features present is considered better than a more complex model. This is in part because of the ease of computation, but also because it reduces the possibilities of overfitting the model as well as including highly correlated predictors.

This process begins by gathering all of the measures that one might want to include in the model. Once this list, of size n, has been created, then each measure alone is used to calculate the accuracy of the algorithm. The most accurate measure is added to the final measure list. Then that single most accurate measure is paired with each of the remaining measures to create $n-1$ lists of two measures. Each of the new two-measure lists is used to calculate the model accuracy and the most accurate is added to the final measure list. This process is repeated until all of the measures are in one list. In the end, there should be n most accurate lists ranging in size from one to n. The overall most accurate is determined

to be the best measure list for that trial. If multiple combinations of features produced the same accuracy, the combination with the least features is deemed best. Multiple trials are then completed, determining a best measure list for each trial. The number of trials completed is designated as $tr_{\#}$. Trial design is discussed further in Section 12.2. Once measure lists have been created for all of the trials, scoring is performed. Classification accuracy, and other performance statistics, are averaged across all of the trials.

12.1.1 Macro- v Micro-Scoring

The difference between micro-scoring (m-CS) and macro-scoring (M-CS) occurs after all the trials have completed the build stage. Thus there are $tr_{\#}$ unique measure lists, one for each trial. Under macro-scoring, each trial uses their unique, customized measure list to classify the test graphs, resulting in macro-lists.

Under micro-scoring, one measure list is created from the $tr_{\#}$ lists. The micro-scoring list is created in several steps. First, we look to see if any graph measures occur in all of the macro-lists. If so, these measures are automatically included into the micro-list. Next, we look for any patterns within the macro-lists. These patterns can be anything from length of the list to obvious groupings between the number of time measures appear in macro-lists or combinations of measures that always appear together (or never do). Creation of the micro-list requires more judgment than for the macro-list, thus there is more room for potential human error.

12.1.2 Importance-Scoring

The third way to design the measure list does not involve the build stage. Importance-scoring is run by declaring the measures that are most important to the classification problem under investigation. Mathematical proof of the accuracy of this method is discussed in Section 13.3.7.

12.2 Data

The purpose of this chapter is to explain the three ways that the measure list for cross scoring can be determined. Therefore, we present this explanation through the use of a

simplified example. Only a fraction of the available data, both in terms of model graphs and graph measure, are used. The graph measures considered in the example experiment performed and displayed in the results are the number of nodes in the graph, the number of edges, graph density, transitivity, and assortativity.

The setup of the model graphs differs from the setups seen in previous chapters. We still work with the 1000 model graphs of each of the nine types, however these graphs are randomly split into four groups, each containing 250 graphs of each type. The groups are then assigned to one of three categories: build, test, and comparison (Figure 12.1). Two groups at a time make up the comparison category.

Figure 12.1. **Trial design description of model graphs for cross scoring.** The 1000 model graphs of each type are split into four groups, each containing 250 graphs of each type. The groups are then designated as build, test or comparison.

The graphs in the comparison group are used to create the measure values to which the graph to be classified is compared. The build group consist of the graphs that will be classified during the creation of the measure list. With four groups of graphs, we have the potential to run twelve trials. For this example, we only consider four trials (Table 12.1). The full number of trials is considered in the next chapter. For this example, however, we only consider four trials.

Table 12.1. Model Graph Groups for Cross-Validation

Trial	Test	Build	Comparison
1	1	2	3, 4
2	2	1	3, 4
3	1	3	2, 4
4	3	1	2, 4
5	1	4	2, 3
6	4	1	2, 3
7	2	3	1, 4
8	3	2	1, 4
9	2	4	1, 3
10	4	2	1, 3
11	3	4	1, 2
12	4	3	1, 2

The trial column just enumerates through the list. The numbers in the test, build, and comparison columns indicate the group of graphs that are acting in each category.

12.3 Results

We ran four different trials allowing each group to take part in the build stage once. As was mentioned, we began by calculating the classification accuracy of each graph measure alone. Table 12.2 shows the results for the first trial. Transitivity classified the 2250 test graphs (250 from each of nine types) most accurately at 61.78%. When transitivity and the remaining four measures are used together to classify the graphs, the most accurate combination is density and transitivity (Table 12.3). This combination correctly classified 70.53% of the graphs.

Table 12.2. Trial 1 build stage results at the end of round 1.

Measure	Accuracy
# nodes	47.11%
# edges	29.51%
density	43.60%
transitivity	61.78%
assortativity	42.58%

Table shows the classification accuracies of using each of the five measures alone. The highlighted row has the highest accuracy.

This process continued until all of the measures were used in one single list. Then it was repeated for the remaining three trials. The most accurate measure list of each size for each trial is displayed in Table 12.4. Several features can be noticed from the table. First, transitivity is always the first measure added and its accuracy alone averages 61.92%

Table 12.3. Trial 1 build stage results at the end of round 2.

Measures	Accuracy
# nodes, transitivity	68.49%
# edges, transitivity	53.73%
density, transitivity	70.53%
assortativity, transitivity	60.13%

Table shows the classification accuracies of using each measure in conjunction with the best performing measure from round 1, Table 12.2. The highlighted row has the highest accuracy.

($\pm$1.55) across the four trials. Density or number of nodes is always the second measure added. In the trials where it is not second, it is the last measure added. Assortativity is always the third measure and number of edges is always fourth.

We note also that even when the exact same measure list appears, the accuracy is not the exact same. In some instances it may be the best measure list for one trial, but not for another. Take for instance the first trial measure list of size three. This has an accuracy of 71.73% and is the second best measure list from the trial. The exact same list appears in the third trial, but here the accuracy is up to 73.56% and it is the best measure list. Finally, we note that using more measures is not always better. In two instance, the best measure list contains all of the measures, but in the other two it contains three or four measures. Going from the best measure list to the one containing all of the measures results in about a 1-2% loss in accuracy in these instances.

12.3.1 Macro-lists

The highlighted lists in Table 12.4 are the macro-lists for each of the four trials. Two of the lists are the same and the other two are unique.

12.3.2 Micro-list

The average accuracies and standard deviations for the best measure list of each size are shown in Table 12.5. In this instance, it appears that the average best length of measure list is five features, but it is followed closely behind by the list with three features (72.32% v 72.18%). Since these accuracies are nearly indistinguishable, and both have small standard deviation, it is proposed that two micro lists are considered, one of length three and the

Table 12.4. Most accurate measure lists from build stage results across all rounds for all trials.

Trial	Size	Measures	Accuracy
1	1	transitivity	61.78%
	2	density, transitivity	70.53%
	3	assortativity, density, transitivity	71.73%
	4	# edges, assortativity, density, transitivity	73.47%
	5	# nodes, # edges, assortativity, density, transitivity	71.42%
2	1	transitivity	62.84%
	2	# nodes, transitivity	69.56%
	3	assortativity, # nodes, transitivity	72.09%
	4	# edges, assortativity, # nodes, transitivity	72.67%
	5	density, # edges, assortativity, # nodes, transitivity	73.42%
3	1	transitivity	63.24%
	2	density, transitivity	68.98%
	3	assortativity, density, transitivity	73.56%
	4	# edges, assortativity, density, transitivity	70.13%
	5	# nodes, # edges, assortativity, density, transitivity	72.67%
4	1	transitivity	59.78%
	2	# nodes, transitivity	70.09%
	3	assortativity, # nodes, transitivity	71.33%
	4	# edges, assortativity, # nodes, transitivity	70.62%
	5	density, # edges, assortativity, # nodes, transitivity	71.78%

Table shows the most accurate measure list of each size for all of the trials. Trial number is listed in the fist column, followed by the number of measures, the names of the measures, and the classification accuracy. The most accurate measure list in each trial is the best measure list. It is highlighted.

other of length five. The latter list clearly contains all of the metrics considered, but there are several combinations that can be considered for the list of length three.

Table 12.5. Average classification accuracies by number of measures in most accurate measure list.

# Measures	Accuracy (sd)
1	61.91% (1.55)
2	69.79% (0.67)
3	72.18% (0.97)
4	71.72% (1.60)
5	72.32% (0.90)

Table shows the average classification accuracies and standard deviations for each size measure list across the four trials. Results are derived from the accuracies reported in Table 12.4.

The results in Table 12.4 show two different combinations of size length three. The first is assortativity, density, and transitivity. The second replaces density with number of nodes. Both of these combinations occur twice. The average accuracy of the former list is 72.65%. The average accuracy of the latter list is 71.71%. Thus, we choose the first combination of length three to be the micro-list. Thus the two micro-lists can be seen in Table 12.6.

Table 12.6. Micro-lists for cross scoring build stage example.

3	assortativity, density, transitivity
5	density, # edges, assortativity, # nodes, transitivity

Table shows the two micro-lists. The first number is the number of measures in the micro-list.

12.3.3 Importance-Scoring

Importance-scoring involves choosing measures not mathematically, but by relevance to the problem at hand. If we randomly choose four measure, such as betweenness centrality, closeness centrality, number of nodes, and average clustering coefficient, we still achieve an average accuracy of 71.78% (±1.23). Thus, logically picking measures can result in acceptable accuracies.

12.4 Discussion

In this chapter we expanded upon the cross scoring algorithm discussed in the previous chapter. In Chapter **??**, we designed the structure of the algorithm using two measure lists that were generated without being mathematically tailored to the algorithm. Here, we explained the three ways in which we can determine the measure list through an example problem. We used four trials, each classifying 250 graphs from nine model types, and five graph measures. Measures were assessed for accuracy and the most accurate lists of lengths one to five were created. Then the most accurate list of these five was declared the best measure list for the trial. This procedure was run independently for each trial.

Across the four trials, transitivity was always the first measure added. Going back to the results from Chapter 4 (Figure 4.6, Table 4.8) it is clear that assortativity is one of the few measures in which each model graph type has a unique value. This is not true of density, number of nodes, or number of edges (Figure 4.1, Table 4.1). Assortativity does not have as many overlaps as the previous three measures, but it has more than transitivity.

We saw two trials using different groups of graphs for build graphs and comparison graphs arrive at the same best measure list. Even though they had the same best measure list, they did not have the same accuracy. Trial 2 recorded an accuracy of 73.42% and trial 4 had 71.78%. This signifies that despite the fact that the same growth mechanism was

used to build all of the graphs of each type, there are some definite differences between each model graph, even within a model type.

In addition to the fact that two different trials arrived at the same best measure list, many of the measures other than transitivity were also always added in the same order. This could be due to a number of factors. It could be the small number of trials that were run, the reduced number of graph measures used for the comparison, or some combination of both. Due to the setup, it is not surprising that we see so many similarities, though we would not expect 50% of the graphs to match if a complete set of graph measures were used or significantly more trials were run.

The three methods for determining the measure list all have different applications. Macro-scoring can be useful no matter the number of measures used or trials run. It requires no human intervention. Micro-scoring, on the other hand, can be difficult to manage if there are too many measure lists to consider. This can be exacerbated if numerous graph measures are considered and if there are not clear favorites across the trials. Micro-scoring, however, does make it easy to compare results across trials because the graphs are all classified using the same list. Importance-scoring is most useful when only certain measures are important to the problem and one is willing to potentially lose a bit of accuracy in exchange for not losing focus.

Overall, we have shown that multiple different lists of measures can achieve acceptable classification accuracies. By showing this, we have also confirmed that the design of the algorithm was not accidentaly tailored to the lists used to help design it. Thus, we can safely go forward and apply this classifier to the full classification problem.

Chapter 13

Applying the Cross Scoring Algorithm

At this point, the structure of the cross scoring algorithm has been tested and the methods for determining the measure lists for classification have been explained. In this chapter, we assess the classifier's ability to differentiate between different model graph types. In addition, we apply the method to the *S. cerevisiae* PPI network. We conclude this chapter with a comparison of this new classifier to those previously discussed in this dissertation: DDD, CC, RGF (C), and GDD V3.

13.1 Methods

As previously mentioned, there are two stages to the cross scoring classification process: the build stage (discussed in Chapter 12) and the test stage. In the build stage, the best measure lists are determined. In the example used in Chapter 12 only five measures were considered. Here we look at a total of eighteen measures. Those considered as possible predictors are: number of edges, number of nodes, number of triangles, ASPL, assortativity, average clustering coefficient, average degree, average neighbor degree, betweenness centrality, closeness centrality, degree centrality, density, diameter, eigenvector centrality, maximum degree, proportion of nodes in the giant component, radius, and transitivity. Only 171 of the 262,143 possible combinations of measures were tested because of the ways the lists are built, one measure at a time. The best measure lists used to determine the model accuracy are calculated using macro-scoring, micro-scoring, and importance-scoring.

Once the best measure lists have been selected, the cross scoring test stage begins. In this stage, the test graphs, which were not used to choose the measure lists, are classified. The results from all of the trials are combined, resulting in a single average accuracy. The classifier is evaluated by this accuracy, as well as the other performance statistics: PPV, NPV, sensitivity, specificity, and both F-measures. If adequate accuracy is achieved, the *S. cerevisiae* PPI network is classified and the results are interpreted using Bayes theorem.

13.2 Data

The same 9000 model graphs that have been previously utilized were used for classification by the CS algorithm. The graphs are randomly split into four groups, each containing 250 graphs of each type. The groups are then assigned to one of three categories: build, test, and comparison (Figure 12.1). Two groups at a time make up the comparison category.

Each group is given a turn in each position. Two groups at a time make up the comparison graphs (Table 12.1). This results in twelve trials and 3000 graph tests per model type for a total of 27,000 graph tests.

13.3 Results

13.3.1 Measure Selection

Eighteen graph measures were considered for insertion into the cross scoring model. Of these, three appeared in all of the twelve trials' best measure lists (macro-lists): assortativity, average degree, and average neighbor degree (Figure 13.1). It is interesting to note that all of these measures are representations of the way nodes are connected to each other. Three other measures appeared in a macro-list nine or more times: number of nodes (9), betweenness centrality (10), and closeness centrality (11). Three features did not appear in any list: number of triangles, diameter, and radius. The remaining measures all appeared between one and six times. The median number of appearances a measure made was 4.5 with an IQR of 8.5. The mean number of appearances is 5.33 with a standard deviation of 4.56. The distribution can be seen in Figure 13.2. The histogram has a U-shape, with the most extreme values seen most often.

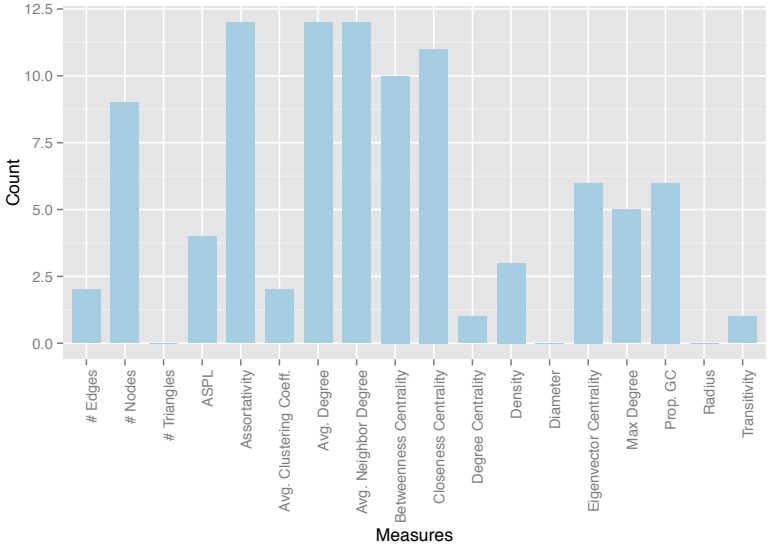


Figure 13.1. **Counts for measure appearance in macro-lists.**

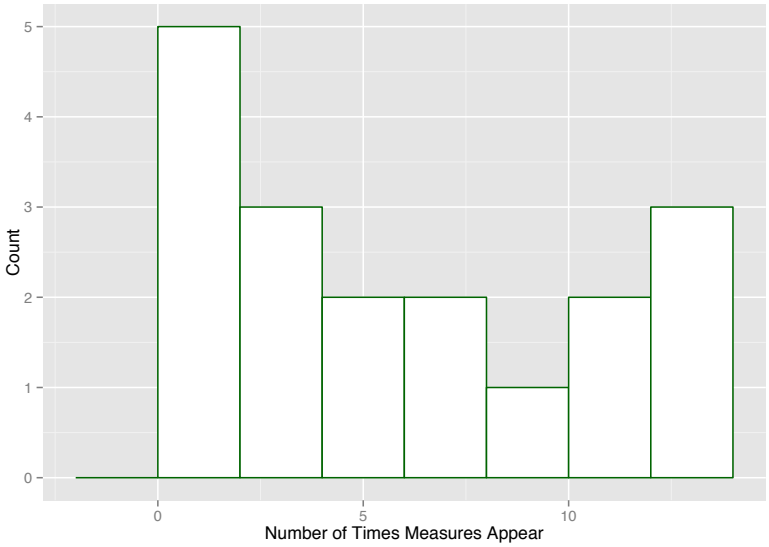


Figure 13.2. **Histogram of number of macro-lists a measure appears in.** The x-axis is the number of macro-lists a measure appears in. The y-axis is the number of measures that appear x times.

13.3.2 Macro-Lists

Twelve trials were run resulting in twelve macro-lists. The majority of measure lists, six out of the twelve, had nine measures in them. Four out of twelve had six measures. Of the remaining two trials, one had seven measures and the other ten.

In the previous section it was mentioned that assortativity, average degree, and average neighbor degree appeared in every model (Figure 13.1). Since these measures are in all of the models, it might be expected that they are usually among the first added. That is true for average neighbor degree and average degree. The former measure is the first added to the list in eleven out of the twelve trials while the latter measure is second in those same eleven (Figure 13.3). One trial resulted in transitivity being added first, thus pushing the previously mentioned measures down in sequence. Assortativity does not respond in the same way as the two other measures. On average, assortativity is the sixth measure to be added to the measure list. This means that in some instances, it was the last measure added to the best measure list.

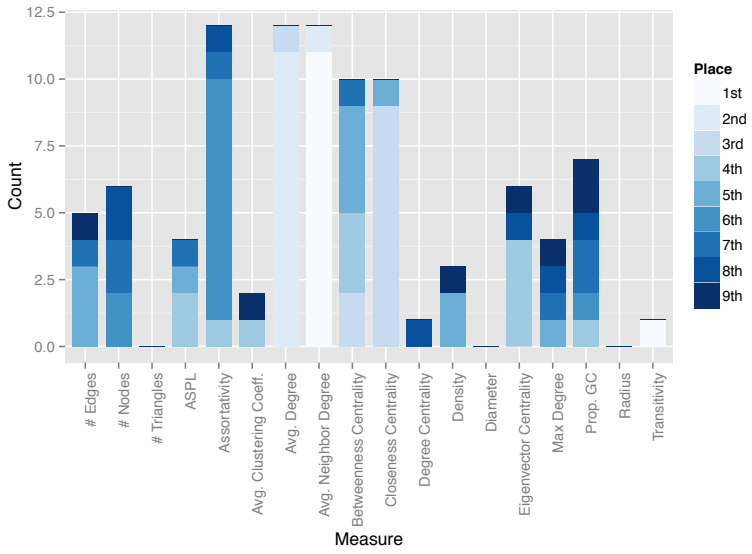


Figure 13.3. **Position each measure is added to the macro-list.** The different colors indicate the position that the measure was added to the list. Lighter colors indicate that the measure was added earlier than dark colors.

The choice of the best measure list for each trial was dictated by the percentage of model graphs accurately classified. This was the only statistic considered in order to at-

tempt to simplify an already extremely complex problem. Figure 13.4 shows the accuracy of each of the twelve trials plotted against the number of measures in the model. Measure lists with only a small number of measures, one or two, performed poorly. By the time there were five measures in the list, however, changes in accuracy are very small, nearly negligible. After five measures are in the list, the increase in accuracy is not smooth, nor is it guaranteed. More measures in the list does not dictate an increase in accuracy. In fact, after approximately ten measures are in the list, accuracy begins to decrease. Fourteen measure lists have about the same accuracy as three measure ones.

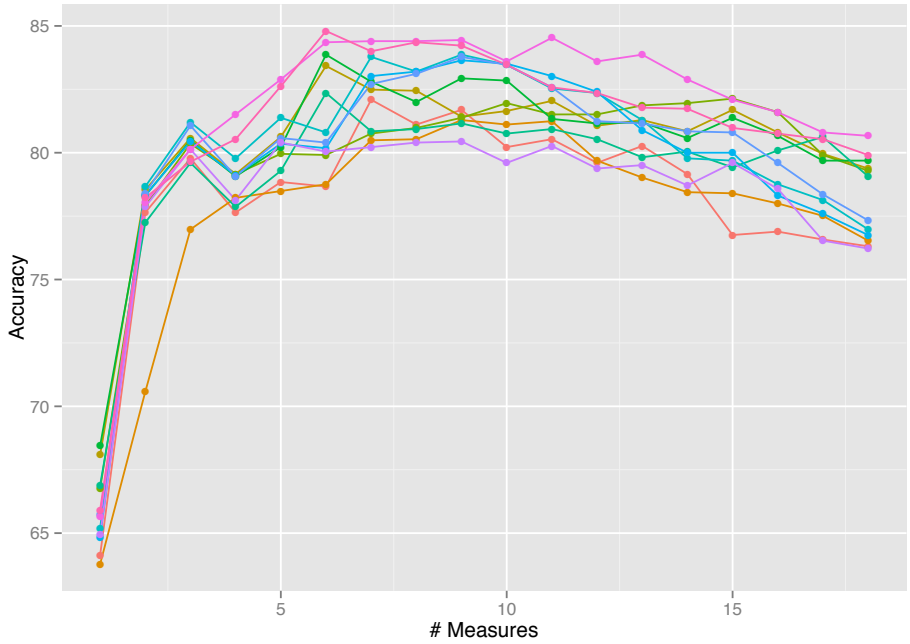


Figure 13.4. **Cross scoring build stage accuracy by trial and number of features in the best measure list.** The x-axis is the number of measures in the best measure list. The best measure list is the most accurate list of size x. The y-axis proves the accuracy. Each line represents one of the twelve trials.

The average accuracy for the best model across the twelve trials in the build stage is 83% ($\pm$1.34). The lowest accuracy is 80.44% and the highest is 84.80%. All of these trials clearly have the potential to out-perform all of the other classification algorithms considered.

The measures in each of the twelve macro-lists are displayed in Table 13.1. The macro-lists for Trials 4 and 5, as well as Trials 7, 8 and 10, are the same. Trials 4 and

5 share only one of the two groups in the comparison graphs. Trial 4 comparison graphs are made of group 2 and group 4, while trial 5 has group 3 instead of group 4. Trials 7, 8 and 10 also share one of the two groups making up the comparison graph. Trials 7 and 8 contain the exact same comparison groups, 1 and 4. Trial 10 has comparison graphs from groups 1 and 3. The features in the measure lists for Trials 4 and 5 are not added in the exact same order. The measure lists for Trials 7, 8, and 10 are exactly the same in terms of order added as well as content.

Table 13.1. Macro-lists.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10	Trial 11	Trial 12
# Edges		x	x									
# Nodes			x	x	x	x	x	x	x	x	x	
# Triangles												
ASPL		x	x						x		x	
Assortativity	x	x	x	x	x	x	x	x	x	x	x	x
Avg. Clustering Coefficient		x	x									
Avg. Degree	x	x	x	x	x	x	x	x	x	x	x	x
Avg. Neighbor Degree	x	x	x	x	x	x	x	x	x	x	x	x
Betweenness Centrality	x	x		x	x	x	x	x	x	x		x
Closeness Centrality	x		x	x	x	x	x	x	x	x		x
Degree Centrality		x										
Density		x	x								x	
Diameter												
Eigenvector Centrality				x	x	x			x		x	x
Maximum Degree				x	x				x			
Prop. GC	x			x	x	x			x			x
Radius												
Transitivity											x	

Table shows which measures appear in each trial's macro-list. Trials 4 and 5 have the exact same macro-list, as do Trials 7,8, and 10.

13.3.3 Macro-Scoring Performance

In five of the twelve trials test accuracy actually exceeded build accuracy (Table 13.2). These trials are written in boldface in the table. In six trials, the differences in accuracies were within 1% of each other. These trials are annotated with an asterisk in the table. Only two of these six overlap with the four trials that presented with higher than expected accuracy. All of the trials had accuracy values within 5% of the anticipated value.

Model classification for all trial results aggregated can be seen in Table 13.3. It is important to note that this table shows the percent of graphs classified into each category,

Table 13.2. Comparison of macro-list accuracy from test and build stages.

	Accuracy	
Trial	**Test Stage**	**Build Stage**
1	83.33	84.44
2	83.87	84.80
3	82.84	83.78
4	**81.69**	**80.44**
5*	83.64	83.87
6*	**83.87**	**83.64**
7*	82.36	83.87
8	**83.87**	**82.36**
9*	83.11	83.42
10*	**82.27**	**81.96**
11	**83.11**	**82.09**
12*	80.71	81.29

The lines in boldface have higher accuracy in the build stage than in the test stage. The lines indicated with an asterick (*) have values within 1% of each other.

not the absolute number of graphs. Across the twelve trials, the average accuracy is 82.9%. Of the nine model types, three were classified accurately 100% of the time: GEO, LPA, and RDG. Three models were very close to 100% accuracy. RDS was accurately classified 99.93% of the time, SMW was 96% and STI was 99.87%. The remaining three models were slightly less accurate. AGV was correctly classified 67.3% of the time, while DMC was 30.73% and DMR was 49.2%.

Table 13.3. Classification accuracy of macro-scoring.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
Actual Class	AGV	67.3	-	-	-	32.47	-	0.03	0.2	-
	DMC	14.57	30.73	34.43	6.77	4.27	2.1	4.0	1.47	1.67
	DMR	8.73	5.67	49.2	1.67	3.53	9.33	11.7	5.67	4.5
	GEO	-	-	-	100	-	-	-	-	-
	LPA	-	-	-	-	100	-	-	-	-
	RDG	-	-	-	-	-	100	-	-	-
	RDS	0.03	-	-	-	-	-	99.93	0.03	-
	SMW	0.03	-	-	-	-	-	4	96	-
	STI	-	-	0.13	-	-		-	-	99.87

The values presented are the percent of model graphs classified into each category by macro-scoring. The percentages are aggregated over the twelve trials.

With the exception of DMC and DMR graphs, misclassified graphs seem to fall neatly into one category. For instance, the majority of misclassified AGV graphs are classified as LPA. Significantly less than 1% of those misclassified graph fall in RDS and SMW.

Misclassified DMC and DMR graphs are placed into every category. A total of 30.73%, or 922, DMC graphs were classified correctly. The most common misclassification category for DMC is DMR. More DMC graphs were placed into DMR than were correctly classified, 34.42% or 1033 graphs total. The next closest category, AGV, had less than half of the number of DMC graphs misclassified there than DMR (14.57%). The remaining model types all represent small fractions of the misclassified graphs.

Nearly half of the DMR graph were classified correctly (49.2%). The incorrectly classified models were spread out over the other categories with the second most popular category, RDS, obtaining 11.7% of the graphs.

For the most part, the statistics used to analyze classifier performance show great promise for this version of cross scoring (Table 13.4). Both average and global PPV, NPV, sensitivity, and specificity are all above 0.8. The F-macro is 0.8078 and F-micro is 0.8274. Only three models show values meriting concern when examined closer: AGV, DMC, and DMR. All three have high NPV and specificity, which is to be expected because only one-ninth of the graphs to be classified fall into any one of those categories. Thus even if placed randomly, it is more likely for the classifier to say correctly that a graph is not AGV (or DMC or DMR) than it is to incorrectly say that it is. Unfortunately, the low PPV and even lower sensitivity values indicate that the model is also not good at determining which models fit into any of those three categories.

Table 13.4. Macro-scoring analysis of performance.

Model	**PPV**	**NPV**	**Sensitivity**	**Specificity**	**F-measure**
AGV	0.7423	0.9596	0.673	0.9708	0.706
DMC	0.8442	0.9198	0.3073	0.9929	0.4506
DMR	0.5874	0.9378	0.492	0.957	0.5355
GEO	0.9222	1.0	1.0	0.9895	0.9595
LPA	0.7129	1.0	1.0	0.9497	0.8324
RDG	0.8974	1.0	1.0	0.9857	0.9459
RDS	0.8974	0.9999	0.9993	0.9803	0.9267
SMW	0.9287	0.995	0.96	0.9808	0.9441
STI	0.9418	0.9998	0.9987	0.9923	0.9694
Average	0.8268	0.9791	0.8256	0.9788	0.8078
Global	0.8292	0.9999	0.8256	0.9788	0.8274

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the macro-scoring aggregated across the twelve trials. This version has the edited algorithm structure and the scaling step was removed. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

In Figure 13.5, we do not see all of the groups that we have become accustomed to seeing. Instead, only Groups 1 and 4 are represented. The former group refers to graphs that are correctly classified most of the time but are also a popular choice for the misclassification of other graphs. Model types in this group are GEO, LPA, RDG, RDS, SMW, and STI. Group 4 is graphs that are classified accurately a moderate amount of the time and used as an incorrect choice some of the time, though they are not as popular as those in Group 1. Model graphs in this group are: AGV, DMC, and DMR. These group classifications are very clearly defined and match easily to the classification accuracies seen in Table 13.3.

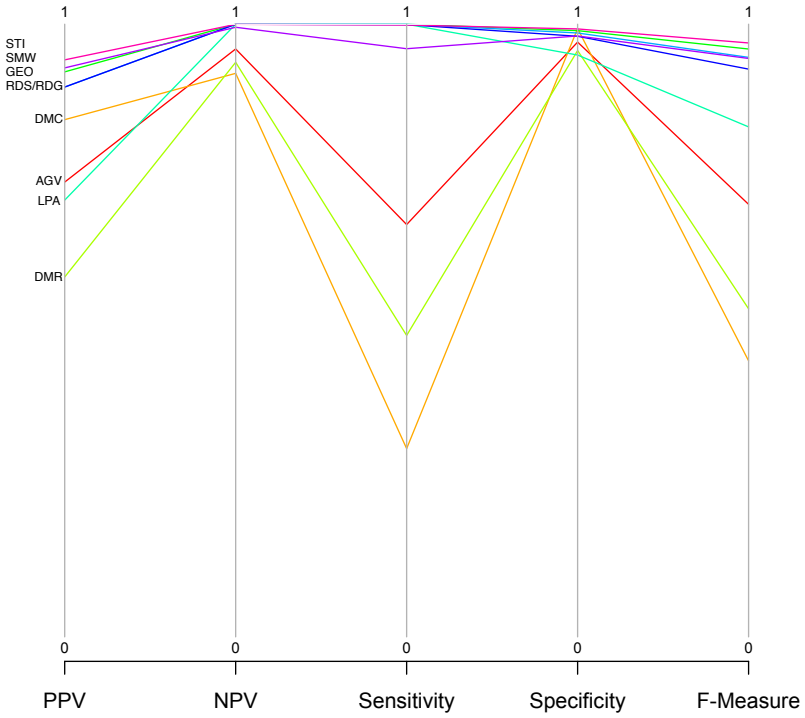


Figure 13.5. **Parallel coordinate representation of the macro-scoring performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated by aggregating the result from the twelve trials. Each trial classifies 100 models of each of the nine model graph types.

13.3.4 Micro-Scoring Performance

In micro-scoring, instead of using the macro-lists directly, we use the number of appearances that each measure makes to determine generic measures lists, or micro-lists, for use with all of the cross scoring trials. We chose to try lists of lengths three, six and nine. Length three was chosen because there are three measures that appeared in all of the lists. List lengths of size six and nine were chosen because Figure 13.1 shows distinct breaks in the number of occurrences at these points.

Table 13.5. Comparison of macro and micro-scoring classification accuracy.

Trial	**Macro-List**	**Micro-3**	**Micro-6**	**Micro-9**
1	83.33	78.12	84.76	83.47
2	83.87	78.93	84.0	83.16
3	82.84	77.82	79.51	82.84
4	81.69	79.69	80.84	83.79
5	83.64	78.13	79.33	83.64
6	83.87	79.2	80.49	83.87
7	82.36	76.89	82.36	80.4
8	83.87	78.8	83.87	81.24
9	83.11	77.73	83.11	81.24
10	82.27	77.82	83.42	81.6
11	83.11	78.22	79.51	81.2
12	80.71	76.53	78.8	80.44
Average	82.9	78.16	81.67	82.27

Micro-3 is the micro-scoring list with three measures. Micro-6 is the list with six measures and micro-9 is the list with nine measures.

The measure list of length three consists of average degree, average neighbor degree, and assortativity. The list of length six also includes the number of nodes in the graph as well as betweenness and closeness centrality. Finally, for the list of length nine, eigenvector centrality, the proportion of nodes in the giant component, and the maximum degree of the graph were added.

Table 13.5 shows the classification accuracy for each trial for macro-lists along with the three micro-lists. Accuracy ranges varied more across the tests using the micro-list than the macro-lists. For the top three measures model, *micro-3*, all of the accuracies were significantly lower than the build accuracies, ranging from 76.53 to 79.69. This is an average of 4.7% smaller than the macro-list accuracies. The top six measures model, *micro-6*, ranged from 78.8 to 84.76% accuracy. This 84.76% accuracy is the highest accuracy achieved across

all of the trials using any measure list. On average, however, accuracies were 1.2% smaller than the macro-list accuracies.

Finally, the model based on the top nine measures, *micro-9*, ranged from 80.4 to 83.87%, almost indistinguishable from the custom model range of 80.71 to 83.87%. The difference in accuracy compared to the macro-lists is less than 1%. In Figure 13.6, we see a comparison of how the measure lists are ranked when compared to each other. The model using the macro-list is the most accurate two-thirds of the time, for a total of eight of the trials. The micro-3 list is always the least accurate. The micro-9 makes appearances in all places except fourth. Since this measure list creates a model with just over 82% accuracy, barely indistinguishable from the custom model, we further evaluate its performance.

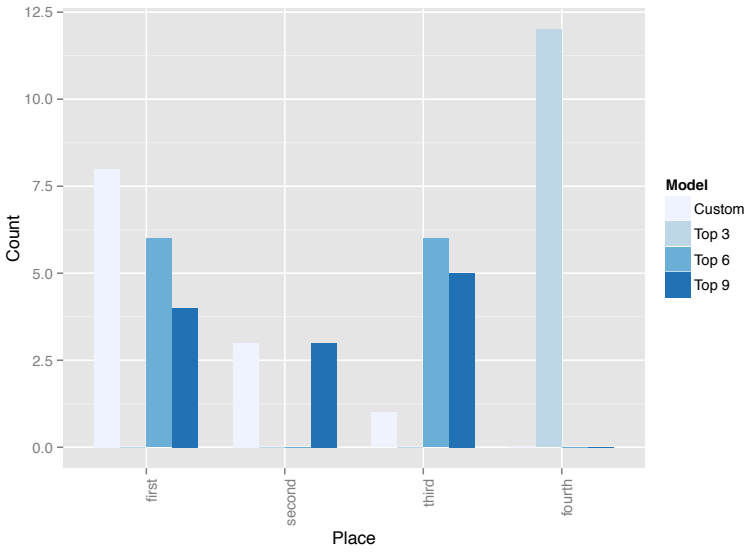


Figure 13.6. **Comparison of accuracy for macro-scoring and three version of micro-scoring across twelve trials.** The x-axis, *Ranking*, indicates where the model ranks in terms of accuracy compared to the others.

Table 13.6 shows the classification distribution of each model type across all of the trials. Just like the corresponding table in the previous section (Table 13.3), this table shows the percentage of graphs that fall into each category, not the raw number. Three graphs were classified accurately 100% of the time: GEO, RDG, and RDS. Three more were classified accurately nearly 100% of the time: LPA at 99.87% and both SMW and STI at 99.97%. Using this classification algorithm, misclassified graphs of nearly all model types

fell into only one category. The exceptions to this are DMC and DMR, both of which were classified at some point into all nine categories.

Only 21.53% of DMC graphs were classified correctly. The most common choice for incorreclty classified DMC graphs was DMR (40%). GEO was a slightly less popular choice with 25% of DMC graphs classified into that categoru. Just over 50% of the DMR graphs were classified correctly. The main grouping of misclassified DMR graphs were placed into RDG (23.47%). The remaining graphs were spread evenly between the other categories.

Table 13.6. Classification accuracy of micro-scoring with nine measures.

		Predicted Class								
		AGV	DMC	DMR	GEO	LPA	RDG	RDS	SMW	STI
Actual Class	AGV	68.33	-	-	-	31.67	-	-	-	-
	DMC	5.07	21.53	40.0	25.07	2.77	2.77	0.9	1.17	0.73
	DMR	4.47	5.3	50.73	5.47	2.73	23.47	1.7	1.13	5.0
	GEO	-	-	-	100	-	-	-	-	-
	LPA	0.13	-	-	-	99.87	-	-	-	-
	RDG	-	-	-	-	-	100	-	-	-
	RDS	-	-	-	-	-	-	100	-	-
	SMW	0.03	-	-	-	-	-	0.03	99.97	-
	STI	-	-	0.03	-	-		-	-	99.97

The values presented are the percent of model graphs classified into each category by macro-scoring. The percentages are aggregated over the twelve trials.

The statistics used to analysis the performance of micro-scoring all show great promise (Table 13.7). Average PPV, NPV, sensitivity, and specificity are all above 0.8 with specificity and NPV being higher than PPV and sensitivity. This indicates that models are better determining what graphs are not than what they are. The F-macro for this method is 0.7984 while the F-micro is 0.8231. All of the NPV and specificity values are very high, however several PPV values and sensitivities are lower than desired. This is particularly true of the AGV, DMC, and DMR sensitivities (0.68 v 0.22 v 0.51). These are low because many of these graphs were not classified correctly. In fact, the majority of DMC graphs were not classified correctly. The only one of the three with a very low PPV is DMR. This occurs because so many of the DMC graphs were incorrectly identified as DMR.

Figure 13.7 shows a visual representation of the statistics used to analyze the classifier's performance. We see only two groups in this visual, Group 1 and Group 4. AGV,

Table 13.7. Micro-scoring with nine measures analysis of performance.

Model	PPV	NPV	Sensitivity	Specificity	F-measure
AGV	0.8757	0.9615	0.6833	0.9879	0.7678
DMC	0.8025	0.9101	0.2153	0.9934	0.3395
DMR	0.5589	0.9391	0.5073	0.95	0.5319
GEO	0.766	1.0	1.0	0.9618	0.8675
LPA	0.7288	0.9998	0.9987	0.9535	0.8426
RDG	0.7921	1.0	1.0	0.9672	0.8840
RDS	0.9744	1.0	1.0	0.9967	0.9870
SMW	0.9872	1.0	0.9997	0.9984	0.9934
STI	0.9458	1.0	0.9997	0.9928	0.972
Average	0.8257	0.9789	0.8227	0.978	0.7984
Global	0.8235	0.9999	0.8227	0.9780	0.8231

Results are calculated based on the classification of the 100 model graphs from each of the nine model types using the micro-scoring with nine measures aggregated across the twelve trials. This version has the edited algorithm structure and the scaling step was removed. Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. The average F-measure corresponds to the F-macro while the global F-measure corresponds to the F-micro.

DMR, and DMC all fall into Group 4. The remaining model types fall into Group 1.

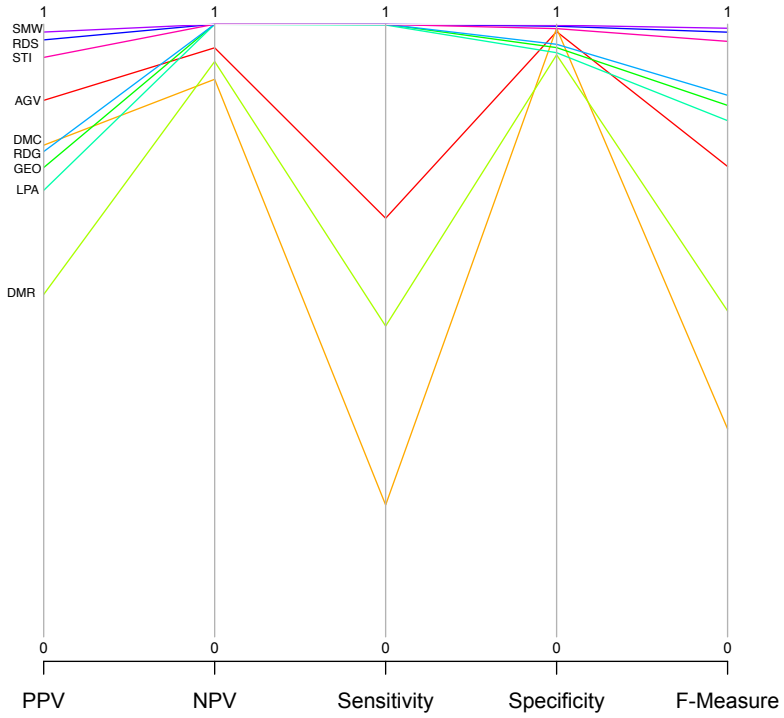


Figure 13.7. **Parallel coordinate representation of the micro-scoring performance statistics.** Statistics used include PPV (positive predictive value), NPV (negative predictive value), sensitivity, specificity, and F-measure. Values are calculated by aggregating the result from the twelve trials. Each trial classifies 100 models of each of the nine model graph types.

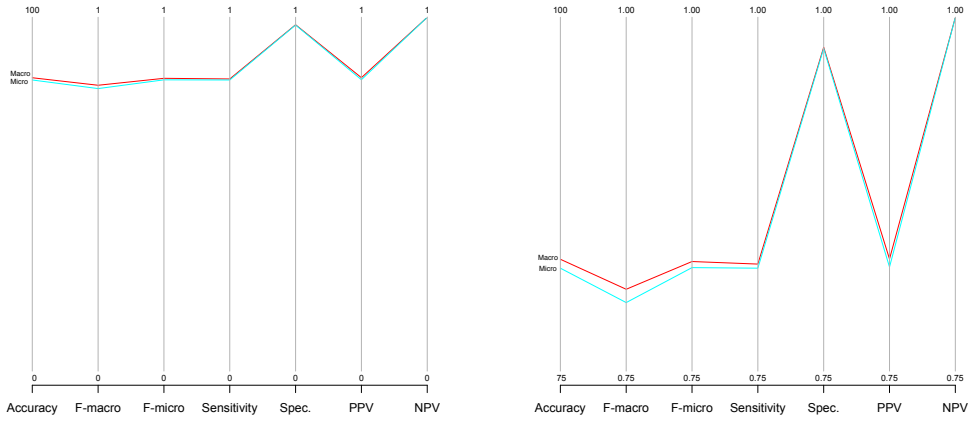
13.3.5 Macro- v Micro-Scoring Results

Macro and micro-scoring both resulted in classification accuracies above 80%. Their performance on other statistics was also very similar. Figure 13.8 shows two views on these performance values. In the image on the left (Figure 13.8a) we see that the lines for both variations almost completely overlap. The macro-scoring procedure does stay just above the micro-scoring, however. In Figure 13.8b, we see a close-up version of the figure on the left. Here we see that the values between the two procedures are virtually identical with the exception of the F-macro and F-micro. For M-CS these values are 0.8078 and 0.8274 respectively. For m-CS they are 0.7984 and 0.8231. The values for macro-scoring are marginally higher for both statistics than the correspondingly micro-scoring values. These values are another way to measure a test's accuracy and thus we can conclude that macro-scoring is more accurate. The main difference in accuracy comes from the classification of the DMC graphs. For macro-scoring 30.73% were correctly classified while in micro-scoring only 21.53% were. This is the only significant difference in classification.

The distributions of the incorrect classifications are very similar between the two cross scoring versions. Incorrectly classified AGV graphs went primarily to LPA, while incorrect DMC graphs went to DMR. Incorrect DMR choices did differ, however. In m-CS these went primary to RDG while in M-CS the split was much more even with a slight majority of graphs were classified as RDS.

It is interesting that both scoring versions produced results allowing graphs to be classified into only Group 1 or Group 4. In fact, the model graphs types were all placed into the same groups no matter which version was used. The elimination of Groups 2 and 3 indicate that no model types can be considered to have low classification accuracy. That is why this is the first time that AGV, DMC, and DMR are classified outside of Group 2 or 3.

Overall, the macro-scoring and micro-scoring with the top nine measures can be used essentially interchangeably. In fact, looking at the percentage of graphs correctly classified in Table 13.5, micro-scoring using the top three or top six measures could also be used without too much loss of information. The choice about which version of micro-scoring to use depends on several factors, most of them relating to time. Extremely large graphs,



(a) (b)

Figure 13.8. **Parallel coordinate comparison of macro-scoring and micro-scoring.** Statistics used include PPV (pos- itive predictive value), NPV (negative predictive value), sensitivity, specificity, and both F- measures. Values presented for sensitivity, specificity, PPV and NPV are the global results of the classification results of the 100 models of each of the nine model graph types aggregated across the twelve trials. **(a):** Full view showing how similar the two variations are. **(b):** Close-up view showing the slight variations.

or a large number of graphs, can make calculations of certain measures, such as centrality measures, very time consuming. We have shown that even using a fraction of the measures available and no centrality measures, such as in micro-3, there is only a 4% loss in accuracy. Under certain circumstances, this loss may be considered negligible when compared to the benefits.

13.3.6 Classification of the S. cerevisiae PPI network

The classification of the *S. cerevisiae* PPI network using both macro and micro-scoring resulted in a classification of AGV. For the macro-scoring, using Bayes theorem, we can determine that there is 67.3% chance that if the *S. cerevisiae* PPI network is AGV then it will be classified as such (Equation 13.1). However, there is a 23.5% chance that even if a model is not AGV, it will still be classified as AGV.

$$\begin{aligned}\Pr(\text{classified as AGV} \mid \text{AGV}) &= \frac{\Pr(\text{AGV} \mid \text{classified as AGV}) \cdot \Pr(\text{classified as AGV})}{\Pr(\text{AGV})} \\ &= \frac{\frac{2019}{2724} \cdot \frac{2724}{27000}}{\frac{3000}{27000}} \\ &= 0.673 \end{aligned} \tag{13.1}$$

$$\begin{aligned}\Pr(\text{classified as AGV} \mid \text{not AGV}) &= \frac{\Pr(\text{not AGV} \mid \text{classified as AGV}) \cdot \Pr(\text{classified as AGV})}{\Pr(\text{not AGV})} \\ &= \frac{\frac{705}{2724} \cdot \frac{2724}{27000}}{\frac{3000}{27000}} \\ &= 0.235 \end{aligned} \tag{13.2}$$

Micro-scoring also found AGV to be the best fit for the empirical model under examination. Once again using Bayes Theorem, the probability that the *S. cerevisiae* PPI network is truly a AGV and was classified as such is essentially the same as was found for macro-scoring (68.3% v 67.3%).

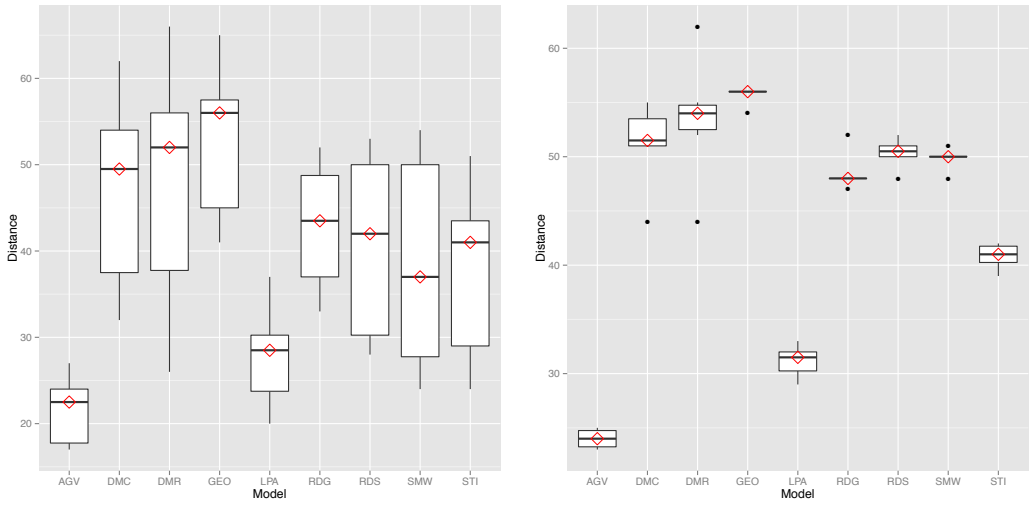
$$\begin{aligned}\Pr(\text{classified as AGV} \mid \text{AGV}) &= \frac{\Pr(\text{AGV} \mid \text{classified as AGV}) \cdot \Pr(\text{classified as AGV})}{\Pr(\text{AGV})} \\ &= \frac{\frac{2050}{2340} \cdot \frac{2340}{27000}}{\frac{3000}{27000}} \\ &= 0.683 \end{aligned} \tag{13.3}$$

$$\begin{aligned}\Pr(\text{classified as AGV} \mid \text{not AGV}) &= \frac{\Pr(\text{not AGV} \mid \text{classified as AGV}) \cdot \Pr(\text{classified as AGV})}{\Pr(\text{not AGV})} \\ &= \frac{\frac{290}{2340} \cdot \frac{2340}{27000}}{\frac{3000}{27000}} \\ &= 0.097 \end{aligned} \tag{13.4}$$

However, there is only a 9.7% chance that the model was classified as AGV and is not actually AGV. Therefore, even though the best fit for the empirical model is the same for both versions of CS, the results from micro-scoring are more reliable.

Figure 13.9 shows the distribution for median distance for each model type across the twelve CS trials. It is clear for both M-CS (Figure 13.9a) and m-CS (Figure 13.9b) that AGV distances are dramatically smaller than any other model type. The second best fit for both versions is LPA. It is interesting that in macro-scoring, the range of distances achieved by each model graph type is larger. This is most likely due to the use of different measures used in the comparisons across the twelve trials. These different measures would highlight different features, while in micro-scoring the same measures were used in each of the trials. The fact that the same model type was deemed the best fit for the *S. cerevisiae* PPI network by both versions of the CS algorithm lends further credibility to the methodology.

Comparing the overall rankings of the model graph by macro- and micro-scoring using Kendall's W results in a value of 15.2 (Table 13.8). This value has a corresponding p-value greater than 0.05, thus we can say that the difference between these lists is not statistically significant The lists are identical for the top two and the bottom three model types. The graphs with middle rankings, SMW, STI, RDS, and RDG, appear in different orders between the two lists, however these models never move more than two places in either direction.



(a) (b)

Figure 13.9. **Comparison of *S. cerevisiae* PPI network classification by macro and micro-scoring.** Each figure shows the results of comparing the empirical S. cerevisiae PPI network against the 1000 model graphs of each of the nine types. Results are displayed such that the red diamond across the middle line is the median distance, the ends of the box are the first and third quartiles respectively, and the remaining lines and points represent the outlying distances. **(a):** The results using macro-scoring. AGV is the best fit. **(b):** The results using micro-scoring. AGV is the best fit.

Table 13.8. Ordered rankings of the model graphs based on fit for S. cerevisiae PPI network using the original graphlet degree distribution and reformulated version 3.

	Macro	Micro-9
1	AGV	AGV
2	LPA	LPA
3	**SMW**	**STI**
4	**STI**	**RDG**
5	**RDS**	**SMW**
6	**RDG**	**RDS**
7	DMC	DMC
8	DMR	DMR
9	GEO	GEO

Rankings of model graphs are determined based on the median distance of the S. cerevisiae PPI network to each model graphs of the given type. The median smallest distance is ranked first. Models in bold show up in different places across the two lists. Items not in bold do not change position when the reformulated graphlet degree distribution version 3 is used in place of the original.

13.3.7 Importance-Scoring

In Chapter 12 we touched upon the idea of importance-scoring. This means that instead of mathematically determining the measure list to use when classifying the graphs, we instead determine the measure list by choosing measures that are of great importance to the problem. In Chapter 12, we gave an example that illustrated how randomly choosing four measures resulted in an accuracy of 71.78%. Now we show that acceptable accuracy can be obtained from any combination of graph measures, provided an appropriate number of them are used.

Table 13.9. Proof of importance-scoring's ability to accurately classify graphs.

# of Measures	**% Accuracy (sd)**		
	Best Measure List	Partial Best Measure List	All Combinations
1	65.86 (1.45)	48.26 (10.08)	48.7 (10.05)
2	77.48 (2.21)	65.69 (7.58)	59.65 (8.81)
3	80.03 (1.09)	75.15 (4.40)	66.15 (6.73)
4	79.10 (1.12)	76.80 (2.52)	69.62 (5.11)
5	80.47 (1.34)	77.48 (2.68)	71.71 (4.06)
6	81.46 (2.19)	78.02 (2.61)	73.13 (3.25)
7	82.30 (1.43)	79.83 (2.03)	74.03 (2.73)
8	82.22 (1.44)	80.00 (1.90)	-
9	82.52 (1.42)	80.55 (1.90)	-
10	82.14 (1.45)	80.41 (1.90)	-
11	81.93 (1.20)	80.51 (1.55)	-
12	81.24 (1.30)	80.47 (1.43)	-
13	81.00 (1.27)	79.84 (1.51)	-
14	80.41 (1.34)	79.47 (1.57)	-
15	80.24 (1.60)	79.17 (1.76)	-
16	79.64 (1.52)	79.11 (1.53)	-
17	78.85 (1.58)	78.61 (1.58)	-
18	78.18 (1.63)	78.18 (1.63)	-

The best measure list is the most accurate list with n measures. The values are averaged over the twelve trials. The partial best measure list is the of size n that is composed of the $n-1$ best measure list combined with each of the remaining measures. These values are also averaged over twelve trials. The accuracies presented in the final column are for all lists of each size. These values are not averaged over twelve trials because of the magnitude of the calculations that would require. The values presented are the average percents of graphs accurately classified along with the standard deviation in parentheses.

The first column of Table 13.9 (*# of Measures*) indicates the number of measures in the measure list. The second column, *Best Measure List* shows the average classification accuracy across the twelve trials of the best measure list of each size. This relates back to the Table 12.5 in Chapter 12. The average classification accuracy, using only one measure across the twelve trials in 65.86% ($\pm$1.45%). In eleven of the twelve trials this measure was the average neighbor degree. In the remaining trial it was transitivity. Note that using one

measure, as long as it is the correct one, results in a higher classification accuracy than all of the originally considered classifiers.

The next column in Table 13.9, *Partial Best Measure List*, is slightly different. In the first row, for one measure, it shows the average accuracy for all of the measures individually across the twelve trials, not just the measure with the best accuracy. Interestingly, even choosing one graph measure randomly, the CS algorithm will be more accurate than the DDD. In the second row, the best singular measure is combined with all of the other measures individually to create measure lists of size two. Thus, if average neighbor degree and any other graph measure are combined, they will, on average, have a better classification accuracy than CC or RGF (C). In order to beat the GDD-V1, one only needs to know the two most accurate measures, typically average neighbor degree and average degree, because those two measures combined with any other measure produces an average accuracy of 75.15% ($\pm$4.40%).

The final column in Table 13.9, *All Combinations*, shows the average value for all combinations of size $\in \{1, \ldots, 7\}$. The average is not available for every number of possible measures in the list because of the sheer number of options that this provides. For instance, there are 43,758 possible lists of eight measures from the full list of eighteen measures. For similar reasons, the results presented are only for one trial, not the standard twelve. The final column shows that even if you pick any seven random measures, you will still have a more accurate classifier than all of the other ones presented.

These features of the CS algorithm, imply several things. First, knowledge of even a few of the most accurate features will produce a high level of classification accuracy. Second, any of the features that may be considered most important to the classification process can be combined with other features to create an individualized, accurate classifier. Third, accuracy does decrease as the number of graph measures increases (Table 13.9, Figure 13.4). This happens both for the top measure list and partial top measure list after nine features have been included in the measure list.

13.3.8 Robustness

Being able to correctly classify graphs that contain incorrect or missing data is an important feature of any classification algorithm. Of the original methods considered (DDD, CC, RGF, and both forms of GDD) only the authors of CC (Su *et al.* , 2011) and RGF (Przulj *et al.* , 2004) showed evidence that their classification method is robustness to noise or missing data. Based on previous work into the affect of missing data on graph measures, we expect CS to be essentially unaffected by the inaccuracies in the *S. cerevisiae* PPI network.

Methods for dealing with noisy or incomplete network data are still in their infancy, however there have been several explorations into their affect on network measures (Costenbader & Valente, 2003; Kossinets, 2006; Borgatti *et al.* , 2006; Stomakhin *et al.* , 2011; Bray *et al.* , 2015). Borgatti found that centrality measures for random graphs decline "smoothly and predictably with the amount of error" (Borgatti *et al.* , 2006). Similar results were found by Costenbader who examined centrality measures on empirical networks (Costenbader & Valente, 2003). She found eigenvector centrality to be the most robust of these measures. Results looking at graph-level measures also found a predictable decline (Bray *et al.* , 2015). Since the studies found missing and noisy data to affect measures in a smooth and predictable manner, and since the model graphs used to classify the PPI network are created with noisy data, we speculate that the affect of these inaccuracies is negligible. Because CS is based on graph measures, we therefore expect it to be highly robust to incorrect or missing data.

13.3.9 Comparison of All Classifiers

The cross scoring algorithm was designed because the previously considered classifiers, DDD, CC, RGF (C), and GDD-V3 were all found to be lacking. They lacked accuracy, the ability to take multiple features into consideration, and flexibility. In Table 13.10, the overall classification accuracies for each method are presented along with model specific classification accuracies. From this table it is immediately obvious how much of an improvement the macro- and micro-scoring algorithms are over the previously presented classifiers. The main improvement of the CS over the other methods is that every model graph type has a few graphs classified correctly. Across the four other methods, each has at least one

model type that they were never able to classify correctly. This model type is typically either AGV, DMC, DMR, or some combination of the three. In DDD, no LPA graphs were classified correctly in addition to no DMC. Characteristic curve classified none of the three listed model types correctly. The RGF (C) classified no AGV or DMR graph correctly, and only one SMW. Finally, GDD-V3 did not classify any DMC models correctly.

Table 13.10. Comparison of the classification accuracy of all updated, reformulated, and novel classifiers.

	Classification Accuracy						
Network Type	**DDD**	**CC**	**RGF (C)**	**GDDV1**	**M-CS**	**m-CS**	**Average**
AGV	9	0	0	90	67.3	68.33	**32.4**
DMC	0	0	12	0	30.73	21.53	**10.7**
DMR	8	0	0	20	49.2	50.73	**19.7**
GEO	73	100	100	100	100	100	**95.5**
LPA	0	98	100	90	100	99.87	**83.0**
RDG	100	98	100	100	100	100	**99.7**
RDS	97	100	100	100	99.93	100	**99.4**
SMW	99	76	1	80	96	99.97	**77.0**
STI	25	50	100	100	99.87	99.97	**79.1**
Average	**45**	**58**	**57**	**72**	**82.9**	**82.27**	

The values in the table indicate the percentage of the given model graph that was accurately classified by the classification method. Six classification methods are shown: degree distribution distance (DDD), characteristic curve (CC), corrected relative graphlet frequency (RGF (C)), reformulated graphlet degree distribution version 3 (GDD V3), macro-scoring (M-CS), and micro-scoring (m-CS).

Looking closer into the results of the CS classifier, we see several other differences when compared to the four other classifiers. One thing to consider is that the most popular incorrect choice for both version of CS is DMR. A total of 28% of the incorrectly M-CS classified graphs and 25% m-CS classified graphs were placed into DMR. This is because the majority of DMC models fell into this category. The second most popular choice was AGV for M-CS (19%). In no other model was DMR ever a common choice for misclassifications, nor was AGV.

In Figure 13.10, the average performance statistics for all of the classifiers are shown. The shape of each line appears to follow the same pattern. The highest peak for each method occurs at specificity. There is a large drop into PPV from there and then they shoot back up to NPV. At no point to the lines cross each other. A classifier with a higher value for

one statistic will have higher values for all of the statistics.

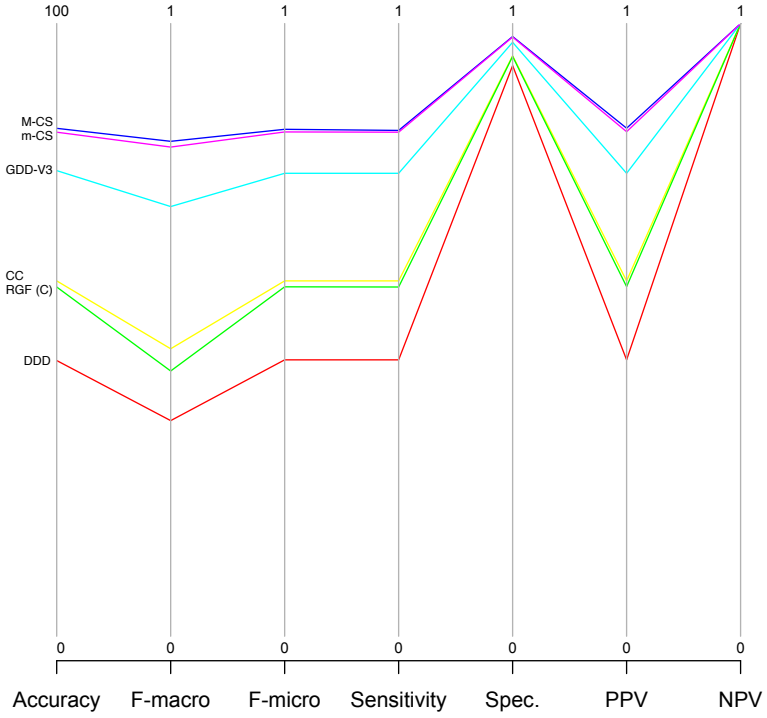


Figure 13.10. Parallel coordinate representation for all classifier performance statistics.

13.4 Discussion

The cross scoring algorithm was created to classify networks based on several important features. It was designed so that it would be customizable, able to fit numerous circumstances without a too significant loss of accuracy. When building the measure lists used to classify the graphs, three measures were never selected into the best model in the build stage. One

of these was the number of triangles in the graph. This is surprising since the top three measures, average degree, average neighbor degree and assortativity, all explain how nodes are connected to each other, just like the number of triangles does. Correspondingly, transitivity was chosen to be in the best model only one time. This is counterintuitive because in Section 4.2, it was clearly shown that most models have drastically different median transitivity values than others. The values range from 0.003 to 0.16 (Table 4.8). This is a 530% increase from the bottom of the range to the top. A measure with such a wide range of values should be a good indicator. The reason that it is clearly not is most likely due to interactions with other measures.

The other two graph measures that were never selected for the best model were radius and diameter. This is not unexpected because in Section 4.2 it was shown that differences between diameter and radius values between models were essentially negligible with the exception of the GEO model graph (Table 4.3). Aside from GEO, diameter values range from 7 to 11 and radius from 5 to 7. This is only a 57% increase across the range for diameter and a 40% increase for radius. GEO had a diameter of 34 and a radius of 19.

The order in which measures were added to the best measure list is also interesting. Assortativity appeared in every model created from the twelve trials, thus this is clearly an influential measure (Figure 13.1). However, it was added to the model in 6^{th} place on average (Figure 13.4). Since many of the measure lists contained only six measures (four models total), this indicates that it was potentially added near last much of the time. In fact it was the last measure added in three trials. This implies that it only works to separate out model types in the presence of other measures. We can speculate that these other measures might be betweenness centrality, number of nodes, or density. In six of the models, assortativity was added immediately after betweenness centrality. In four it was added after the number of nodes and in three of those trials the number of nodes was added immediately after betweenness centrality. In the remaining two trials, assortativity was added after density.

The last thing to consider about the build stage of the model is that the differences between the best model and the second best, and often third best as well, in each trial are extremely small. Often the second best model is less than 0.1% less accurate than the

model created using the best measure list. In some situations it may be optimal to use a shorter, and nearly just as accurate list of measures as the best, in the interest of time and simplicity.

13.4.1 Biological Implications and their Effect on the Cross Scoring Design

The cross scoring algorithm was designed to classify PPI networks. Therefore, the ease with which the results can be interpreted and related to biological events is paramount. The affect of each measure on the classification can be clearly seen, thus an interpretation easily made. This desire stopped us from using two common mathematical techniques. The first technique, singular value decomposition (SVD), is often used to reduce the dimensionality of the data (Stewart, 1993). This can be useful depending on the number of graph measures included, however it does make it more difficult to see the direct affect of each measure on the classification. It can also make it difficult to weight the measures properly, if weighting is applied.

Another analysis technique that was discarded is the use of the support vector machine. This is a supervised learning classification algorithms (Bennett & Campbell, 2000), just like CS. In fact, it works in essentially the way, classifying by feature similarity, and can work with the exact same information. However, its processes work in a black box where the contribution of each measure is less obvious. With CS it is possible to look at the effect each measure has on classification individually, with no extra calculation required.

The classification of the *S. cerevisiae* PPI network as AGV has biological implications. The AGV model was designed to mimic the citation network. The longer a paper (node) has been around, the more opportunity it has to collect citations (edges) (Chen & Redner, 2010). However, at some point the knowledge from older papers becomes common knowledge, or people stop referencing the original source, and the paper stop gaining citations. It has been hypothesized that a similar occurrence happens in the evolution of PPI networks. Proteins which evolved first have more time to gain interactions with other proteins, but at some point this ceases to continue due to evolution (Schüler & Bornberg-Bauer, 2011).

13.4.2 Strengths and Limitations

There are several limitations, or potential points of improvement, for the CS algorithm. First, not every possible model was considered. Out of the 262,143 measure lists possible for consideration, only 171 (0.06%) were tested. The reason for this is efficiency. Testing 262,143 model takes significantly more time. It also has the potential to waste resources. For instance, none of the best models contained diameter or radius. If we tested every model, we would have looked at dozens of models containing these features that were shown to be unimportant. The only downside to this is the possible loss of an important model. We may be missing evidence of highly correlated measures or measures that interact with each other in some way.

Another limitation is that the best model was judged only on accuracy. Depending on the situation it might be beneficial to use a different statistic, or even a combination of them. Using sensitivity along with accuracy or one of the F-measures may produce different results that could give way to better classifiers.

Despite these limitations, cross scoring still has multiple strengths that make it a better classifier than the others considered. It is significantly faster due to the reduced number of graph comparisons that need to be run. In any of the other algorithms, the *S. cerevisiae* PPI network must be directly compared to each comparison graph. This results in 9000 comparisons per test graph. For cross scoring, only 9 comparisons had to be run, because the *S. cerevisiae* PPI network is compared to a *median model graph* of each type. Note that this median model graph is simply a compilation of the median features of the all the graphs of the given type.. In addition, since the test graph is compared to the median model graph, it is less sensitive to differences between graphs of a given model type. This is why DMC and DMR graphs were classified more accurately; extreme features were averaged out.

In addition to the smaller number of comparisons, CS also has fewer calculations in a given comparisons. In the degree distribution distance, there is a calculation for each degree in the graphs. Depending on the size of the graph, this can results in hundreds, or even thousands, of calculations per comparison. In CS, the number of comparisons is equal to the number of measures considered. This feature, combined with the smaller number of

comparisons, creates a faster algorithm. The final strength of CS is that it is customizable. Based on Table 13.9, choosing any seven measures will result in a model with approximately 74% ($\pm$2.73) accuracy. This means that if researchers are interested in a model's ability to mimic only certain real-world features, they can feel confident in their ability to achieve good accuracy using only the features that are important to their situation.

Chapter 14

Summary

Determining the best model graph fit for real-world protein-protein interaction (PPI) networks is an important problem for researchers. The answer has the potential to lead to predictions of unidentified interactions between proteins, the discovery of related protein complexes in different species, and, most importantly, the potential to determine the underlying causes of certain diseases. Numerous methods to determine the growth mechanism of PPI networks have been discussed in the literature, however all of them classified these networks into different model graph categories. One of the goals of this dissertation has been to explain these results. In this chapter, we provide an overview of the work performed as well as highlight the interesting results. We end by presenting several avenues for future research.

14.1 Overview

Different classification methods used different subsets of model graphs when classifying PPI networks. This obviously resulted in a lack of agreement across methods. Therefore, we began this dissertation with the goal of determining whether all of the methods could reach agreement if they used the same subset of model graphs, and, if this agreement was not possible, explaining why. To answer this question we chose five classifiers and nine model graphs to explore. The classifiers were selected because of their frequent occurrence in the literature (Przulj *et al.* , 2004; Przulj, 2007; Su *et al.* , 2011). Three of the methods, relative graphlet frequency, graphlet degree distribution using arithmetic mean, and graphlet degree distribution using geometric mean, are based on small-scale properties. These methods base their comparisons on graphlet countr and the distribution of automorphism orbits between graphs. The other two methods, degree distribution distance and characteristic curve, use large-scale properties, focusing on the overall structure of the graphs.

The model graphs used for the comparisons were accumulated from numerous PPI network classification papers (Przulj *et al.* , 2004; Middendorf *et al.* , 2005; Pržulj & Higham, 2006). Three of these graphs were specifically designed to mimic the protein interactions

seen in the real-world networks: DMC, DMR, and STI. The remaining model graphs (AGV, GEO, LPA, RDG, RDS, and SMW) were chosen because they were repeatedly utilized in PPI network classification analyses.

The specific PPI network under investigation throughout this dissertation is the *S. cerevisiae* PPI network, or baker's yeast. It is composed of 1361 proteins and 3222 interactions. All of the interactions (edges) were identified through multiple studies making this a high-confidence dataset (Gavin *et al.* , 2002). The model graphs were all simulated based on the features of the *S. cerevisiae* PPI network. In all, 1000 graphs were simulated for each of the nine model types.

In Chapter 3, we examined the variation between model graphs created with the same growth mechanism through the use of fifteen graph measures. We found that DMC and DMR graphs displayed substantially more variation within their 1000 graphs than any of the other model types. This is due to them not an requiring an input value for number of edges in the graph. We also found that not all of the SMW graphs, designed to have small-world features, display such features, such as small average shortest path length and high clustering. A similar result is seen in some LPA graphs that do not display the scale-free features that they were designed to have. The large amount of variability and lack of expected features arise from the randomness associated with the graph building algorithms. Instead of choosing to edit the algorithms so that we have the desired graph properties, we insist on keeping the graphs as they are. The reason for this is to stay consistent with the literature. If the algorithms for model graphs were edited, we would not be testing reproducibility of results. In addition, editing the model graph algorithms at this point would make it impossible to compare our classification results to existing literature. In Section 14.2.3, we propose two solutions to deal with these nonconforming graphs.

After examining the variation within each model graph category, we tested how well the growth mechanisms reproduced features of the PPI network under investigation. We mentioned that the model graphs were all based on the *S. cerevisiae* PPI network, thus ideally they will display similar values for many of the measures. Eighteen measures were considered for this section and they can be separated into four categories: size, distance, centrality, and connection (Chapter 4). The findings here showed that it was difficult for

the models to mimic any features outside of the size measures. The size measures were easier to match because all of the growth mechanisms take number of nodes as an input, and most take number of edges as well. LPA graphs matched the most features (50%). This indicates that no model graph type is an obvious best fit for the *S. cerevisiae* PPI network, thus the need for classification algorithms.

Before applying the graph classification algorithms to the *S. cerevisiae* PPI network, we first test the classifiers' abilities to correctly classify known graphs. This step is a key feature in the analysis of any classification method, however it was often neglected in the literature (Przulj *et al.* , 2004; Pržulj & Higham, 2006; Przulj, 2007). We performed two separate tests of this ability. In the first, random graphs were simulated with different probabilities, p, of edge creation (Chapter 6). The classifier had to separate the graphs into groups based on p. In the second, randomly selected model graphs were used as test graphs (Chapter 7). The classifier had to accurately name their class when compared to the remaining model graphs. Results from the first test were optimistic. Only the characteristic curve was unable to classify all of the graph accurately 100% of the time. The second test, however, indicated that classifiers are far from reliable. Accuracy ranged from 48% (DDD) up to only 68% (GDD (A)). These accuracies call into question the results of previous analyses using the same classifiers.

The RGF and the both versions of the GDD are available in a program called GraphCrunch. While performing the classifications using the RGF, it was found that there was a mathematical error in the source code for GraphCrunch version 2 Kuchaiev *et al.* (2011) (Chapter 9). This error was corrected, creating RGF (C). This updated classifier was tested using the same random graph and model graph classification problems. The corrected RGF had the same accuracy as the original, however overall classification was not identical (Table 9.1). It was necessary to show the results of the original RGF classification in order to properly compare the results obtained here to previously obtained results using GraphCrunch.

With their low accuracies, the utilized classification methods leave much to be desired. Therefore, we took a two-pronged approach. We reformulated the most accurate of the five classifiers, GDD (A), and also proposed a new classification method.

Three reformulated versions of the GDD (A) were tested (Chapter 10). Two features of the original algorithm were varied across these new algorithms: scaling and overall structure of the distance. The former feature was included in the original algorithm because researchers found that automorphism orbits with high degrees were often noise. Thus they dealt with this noise by decreasing their effect on the distance. Unfortunately, these high degree nodes are not always noise and scaling simply reduces the reliability of the algorithm. The overall structure of the algorithm was altered to see if it could improve classification results. Of the three reformulated algorithms, the method without scaling and with the altered structure had the best accuracy, 76% (Table 10.10).

The final part of this dissertation discusses or creation of a novel graph classification method: cross scoring (Chapter 13). Cross scoring works by ranking how well, on average, model graphs mimic an assortment of empirical graph measures. Median measure values are calculated from the 1000 model graphs of each type. Then these median values are compared to the empirical value. The model type with the closest value is awarded one point, second closest gets two points, and so on. This process is repeated for all of the measures and the points tallied up. The model type with the lowest score is declared the best fit for the test graph.

The cross scoring method proved to be superior to the other classifiers considered. It is faster because each network to be classified is only compared to a one "median graph" per model type. It also is based on simple mathematical principles, which reduces computation time. In addition, cross scoring is customizable. The measures used to determine the best fitting model type can be edited to directly fit the problem. Finally, cross scoring has substantially better accuracy than the previously considered methods: 82.9% ($\pm$0.98%)).

After all of the research has been completed, we are left with two questions. Which model graph type is the best fit for the *S. cerevisiae* PPI network (Section 14.1.1); and do PPI networks exhibit scale-free properties (Section 14.1.2)?

14.1.1 Which Model Type is the Best Fit for the Saccharomyces cerevisiae PPI Network?

We began this dissertation questioning whether different classification methods would classify the *S. cerevisiae* PPI network into the same category if the same subset of graphs

was used. Throughout this research, the *S. cerevisiae* PPI network was classified by each method and the results analyzed using Bayes theorem. This type of analysis allows us to calculate the probability that a graph was falsely placed into the given category. This enabled us to obtain results despite the overall low classification accuracy of the methods. It also enables us to evaluate the accuracy of those results.

Looking at the final five methods (DDD, CC, RGF (C), GDD V3, and CS), we find that the PPI network was classified as RDG, DMC, AGV, and GEO two times. The Bayesian analysis, which provides us with the probability that the empirical network was misclassified into the given category, indicates that only DDD fails to provide an acceptable level of reliability. For this method, the probability that the network was falsely classified as RDG is 35%. If we remove RDG from the list, we are still left with three different model graph types. This leads us to conclude that even if the same subset of model graphs is used, the methods will still not reach an agreement. This implies that all of the methods use different features to make their classification. Since the CS algorithm employs customizable criteria for classification, along with an acceptably low level probability of misclassification, we conclude that this method provides the best answer: AGV.

14.1.2 Do PPI Networks Exhibit Scale-Free Properties?

One of the big questions surrounding PPI networks is whether or not they have scale-free properties. It has been widely accepted that these networks do indeed possess this property (Jeong *et al.* , 2001; Barabási & Oltvai, 2004; Przulj *et al.* , 2004; Joy *et al.* , 2005; Nacher *et al.* , 2009). However, as research into graph theory has expanded and techniques have become more polished, groups of researchers have begun questioning whether this is really true, though their reasoning has varied. Tanaka *et al.* insists that other researchers are performing incorrect analyses and thus coming to an erroneous conclusion about the PPI networks (Tanaka *et al.* , 2005). Others suggest that because the network is not complete, we do not have enough information to make inferences about the global network (Han *et al.* , 2005; Stumpf *et al.* , 2005; Hakes *et al.* , 2008).

We fall into the latter camp and state that the *S. cerevisiae* PPI network is not a scale-free network based on the results in this dissertation. One of the most common ways

that researchers determine a graph is scale-free through the linear appearance of the degree distribution plotted on a log-log scale. We decided not to use this procedure because of the current controversy about whether this method has a mathematical basis and is being performed correctly (Bollobás & Riordan, 2004). Instead, we based this decision on the results described in Chapter 2. In this chapter, we showed the average shortest path length is not proportional to $\log \log n$. This is a hallmark of a scale-free network (Watts & Strogatz, 1998). In addition, the S-metric is not particularly large. It comes in at 0.54 where a score of one indicates a scale-free network.

14.2 Future Work

The research presented in this dissertation provides several avenues for future work. In this section, we present three of these avenues. The first discusses ways in which the analyses performed could be extended. Next, we discuss how some of the more problematic model graphs might be modified to ease the burden of classification. Finally, we end with a theoretical proposition about whether a growth mechanism is the same as a model type.

14.2.1 Extension of Analyses

In this dissertation, the analysis of the classification of model graphs focused largely on overall accuracies. While general trends of misclassification were mentioned, the brevity with which they were discussed leaves room for expansion. One such expansion involves looking at each model as an individual, as opposed to just one of many. By doing so, we might be able to learn whether the same graph was misclassified multiple times by different classifiers. If this is true, it could direct us to variations within the model graphs that cause them to be misclassified.

Another area where the analysis can be extended involves looking at the differences between the median classification distances. In many instances, examining the difference between distances may results in small values that are not statistically different. Thus modifications to the classification methods might have to be made in order to obtain statistically significant results. One specific example of this is in the DDD classification of the model graphs. A huge percentage of were incorrectly classified as RDG. The reason for this is

unclear. A closer look may reveal that the differences in median distance between the first and second best fit are actually negligible.

14.2.2 Redesign of DMC and DMR Growth Mechanisms

A common theme among the performance of all the classifiers is their difficulty in classifying the DMC and DMR graphs correctly. From the five original classifiers (DDD, CC, RGF, GDD (A), and GDD (G)), only 2.2% and 2.4% of these model graphs were classified correctly (Table 7.13). The corrected RGF classed 12% of the DMC graphs correctly, an improvement over the original 0%. GDD V3 classified 20% of DMR graphs correctly and no DMC. This is an improvement over 10% of DMR graphs. Even the CS algorithm struggled, with about 50% of DMR graphs correctly classified for both M-CS and m-CS. Only 30.73% of DMC were correctly classified for M-CS and only 21.53%. No other graphs were classified so poorly.

The DMC and DMR graphs are particularly difficult to classify due to the extremely varied graphs that are produced by these two growth mechanisms. The extreme variation occurs because of two factors. First, the DMC and DMR growth mechanisms do not take number of edges as an input into their algorithm. Second, all lone nodes are removed from the graphs based on literary precedence (Przulj *et al.* , 2004; Middendorf *et al.* , 2005; Pržulj & Higham, 2006; Przulj, 2007; Su *et al.* , 2011). Since the algorithm is not aiming to have any specific number of edges, and the probability of connection is random, there are instances where very few edges are created thus resulting in most nodes remaining unconnected and becoming eliminated. On the other side, there are instances where the probability of connection is very high, thus most nodes are connected to each other. This creates graphs with hundreds of thousands of edges, despite having no more nodes than any other model type.

It is important to not dismiss the DMC and DMR graphs as potential fits for the *S. cerevisiae* PPI network because these growth mechanisms were specifically designed to mimic the actual growth and interaction patterns of proteins (Sole *et al.* , 2002; Vázquez *et al.* , 2003). Thus we propose a solution. We can create modified DMC and DMR growth mechanism that take the required number of edges as an input value. If such a model could

retain the biologically specific features, namely duplication and mutation, as well as create a more consistent set of graphs, then it has the potential to be the best available model. It should also be easier for the classifiers to work with.

14.2.3 Does the Growth Mechanism Define the Model Graph Type?

In addition to being difficult to classify, the DMC and DMR graphs were the only ones misclassified as every model type. Most misclassified graphs of a single type were placed into one or two categories. This is, once again, due to the large variation between graphs of these types. In fact, when compared to the variation of other model types, their ranges are typically several orders of magnitude larger. The large amount of variation combined with the misclassification into every model type leads us to a proposition: the use of a defined growth mechanism does not guarantee the creation of graphs with the same features. Therefore we suggest that a growth mechanism, or algorithm to build the model graph, is not same as a model graph type. In other words, just because a graph is built by the DMC algorithm, does not make it a DMC graph. Instead, we propose that each graph be given a two part classification: growth mechanism followed by model type. For instance, using macro-scoring, 11.7% of DMR graphs were incorrectly classified as RDS. Thus we might consider these graphs their own distinct model type: DMR-RDS.

Future work in this area involves expanding on the idea of there being a difference between model type and growth mechanism. We need to determine a set of criteria to break up model graphs into more descriptive category based both on creation algorithm as well as graph features.

Chapter A

Table A.1. Network symbols and definitions.

Symbol	Definition	Definition/Equation
$\mathcal{G}$	network or graph	Def. 1.2
$\mathcal{H}$	giant component of G	Page 14
$\mathcal{V}$	the set of nodes (vertices) of a graph	Def. 1.2
$\mathcal{V}_{\mathcal{H}}/\mathcal{V}_{\mathcal{G}}$	proportion of nodes in the giant component	Equation 1.4
$\mathcal{E}$	the set of links (edges) of a graph	Def. 1.2
n	the number of nodes in a graph, $\lvert\mathcal{V}\rvert$	Def. 1.1
m	the number of edges in a graph, $\lvert\mathcal{E}\rvert$	Def. 1.1
$\mathcal{D}(\mathcal{G})$	the density of a graph	Equation 1.3
k, k_i	the degree of a node (i)	Page 14
$\bar{k}$	average degree of a network	Equation 1.1
t_i	number of triangles that node i participates in	Page 15
$\lvert C_p\rvert$	number of cycles of size p	Page 15
$\lvert P_p\rvert$	number of paths of length p	Page 15
C_i	clustering coefficient	Equation 1.5
$\bar{C}$	average clustering coefficient	Equation 1.6
$\mathcal{C}(\mathcal{G})$	global clustering coefficient (transitivity)	Equation 1.7
$e(v)$	eccenticity	Equation 1.8
$\bar{\ell}$	characteristic path length	Equation 1.11
$S(\mathcal{G})$	S-metric (normalized)	Equation 1.24
DC_i	degree centrality	Equation 1.25
CC_i	closeness centrality	Equation 1.26
BC_i	betweenness centrality	Equation 1.27
$\psi_i(i)$	eigenvector centrality	Equations 1.28, 1.29
$r(\mathcal{G})$	assortativity	Equation 1.20

Bibliography

Aigner, Martin. 1969. The uniqueness of the cubic lattice graph. *Journal of Combinatorial Theory*, **6**(3), 282–297.

Albert, Réka. 2005. Scale-free networks in cell biology. *Journal of Cell Science*, **118**(21), 4947–4957.

Albert, Réka, & Barabási, Albert-László. 2002. Statistical mechanics of complex networks. *Reviews of modern physics*, **74**(1), 47.

Aliakbary, Sadegh, Habibi, Jafar, & Movaghar, Ali. 2013. Quantification and Comparison of Network Degree Distributions. *CoRR*, **abs/1307.3625**.

Amaral, Luis A Nunes, Scala, Antonio, Barthelemy, Marc, & Stanley, H Eugene. 2000. Classes of small-world networks. *Proceedings of the national academy of sciences*, **97**(21), 11149–11152.

Barabási, Albert-László, & Albert, Réka. 1999. Emergence of Scaling in Random Networks. *Science*, **286**(5439), 509–512.

Barabási, Albert-László, & Oltvai, Zoltan N. 2004. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, **5**(2), 101–113.

Barabási, Albert-László, Albert, Réka, & Jeong, Hawoong. 2000. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, **281**(1), 69–77.

Barabâsi, Albert-Laszlo, Jeong, Hawoong, Néda, Zoltan, Ravasz, Erzsebet, Schubert, Andras, & Vicsek, Tamas. 2002. Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, **311**(3), 590–614.

Barrat, Alain, & Weigt, Martin. 2000. On the properties of small-world network models. *The European Physical Journal B-Condensed Matter and Complex Systems*, **13**(3), 547–560.

Barrat, Alain, Barthelemy, Marc, Pastor-Satorras, Romualdo, & Vespignani, Alessandro. 2004. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, **101**(11), 3747–3752.

Bavelas, Alex. 1950. Communication patterns in task-oriented groups. *Journal of the acoustical society of America*.

Beichl, Isabel, & Cloteaux, Brian. 2008. Measuring the effectiveness of the s-metric to produce better network models. *Pages 1020–1028 of: Simulation Conference, 2008. WSC 2008. Winter*. IEEE.

Bennett, Kristin P, & Campbell, Colin. 2000. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, **2**(2), 1–13.

Bollobás, Béla. 1998. *Random graphs*. Springer.

Bollobás, Béla, & Riordan, Oliver. 2004. Robustness and vulnerability of scale-free random graphs. *Internet Mathematics*, **1**(1), 1–35.

Bonacich, Phillip. 1987. Power and centrality: A family of measures. *American journal of sociology*, 1170–1182.

Borgatti, Stephen P, Carley, Kathleen M, & Krackhardt, David. 2006. On the robustness of centrality measures under conditions of imperfect data. *Social networks*, **28**(2), 124–136.

Brandes, Ulrik. 2008. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, **30**(2), 136–145.

Bray, Margaret, Hertzberg, Vicki Stover, Elon, Lisa, & Lowery-North, Douglas W. 2015. *Effect of missing data on contact networks: A simulation study*. Unpublished.

Broder, Andrei, Kumar, Ravi, Maghoul, Farzin, Raghavan, Prabhakar, Rajagopalan, Sridhar, Stata, Raymie, Tomkins, Andrew, & Wiener, Janet. 2000. Graph structure in the web. *Computer networks*, **33**(1), 309–320.

Burton, Andrea, Altman, Douglas G, Royston, Patrick, & Holder, Roger L. 2006. The design of simulation studies in medical statistics. *Statistics in medicine*, **25**(24), 4279–4292.

Callaway, Duncan S., Hopcroft, John E., Kleinberg, Jon M., Newman, M. E. J., & Strogatz, Steven H. 2001. Are randomly grown graphs really random? *Phys. Rev. E*, **64**(Sep), 041902.

Chen, P, & Redner, Sidney. 2010. Community structure of the physical review citation network. *Journal of Informetrics*, **4**(3), 278–290.

Cohen, Reuven, & Havlin, Shlomo. 2003. Scale-free networks are ultrasmall. *Physical Review Letters*, **90**(5), 58701.

Cohen, Reuven, Havlin, Shlomo, & Ben-Avraham, Daniel. 2003. Structural properties of scale free networks. *Handbook of graphs and networks*, **4**.

Costenbader, Elizabeth, & Valente, Thomas W. 2003. The stability of centrality measures when networks are sampled. *Social networks*, **25**(4), 283–307.

De Las Rivas, Javier, & Fontanillo, Celia. 2010. Protein–Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks. *PLoS Comput Biol*, **6**(6), e1000807.

Dorogovtsev, Sergey N, Mendes, José Fernando F, & Samukhin, Alexander N. 2000. Structure of growing networks with preferential linking. *Physical review letters*, **85**(21), 4633.

Emmert-Streib, Frank. 2012. Limitations of Gene Duplication Models: Evolution of Modules in Protein Interaction Networks. *PLoS ONE*, **7**(4), e35531.

Erdős, P., & Rényi, A. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci., Ser. A*, **5**, 17–61.

Estrada, Ernesto. 2011. *The Structure of Complex Networks: Theory and Applications*. Oxford University Press.

Faloutsos, Michalis, Faloutsos, Petros, & Faloutsos, Christos. 1999. On power-law relationships of the internet topology. *Pages 251–262 of: ACM SIGCOMM Computer Communication Review*, vol. 29. ACM.

Fiedler, Miroslav. 1973. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, **23**(2), 298–305.

Freeman, Linton C. 1979. Centrality in social networks conceptual clarification. *Social networks*, **1**(3), 215–239.

Fulton, Debra L, Li, Yvonne Y, Laird, Matthew R, Horsman, Benjamin GS, Roche, Fiona M, & Brinkman, Fiona SL. 2006. Improving the specificity of high-throughput ortholog prediction. *BMC bioinformatics*, **7**(1), 270.

Gavin, Anne-Claude, Bosche, Markus, Krause, Roland, & et. al. 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, **415**(6868), 141–147.

Gilbert, Edgar N. 1959. Random graphs. *The Annals of Mathematical Statistics*, 1141–1144.

Hadley, Michael W, McGranaghan, Matt F, Willey, Aaron, Liew, Chun Wai, & Reynolds, Elaine R. 2012. A new measure based on degree distribution that links information theory and network graph analysis. *Neural Systems & Circuits*, **2**(1), 1–15.

Hakes, Luke, Pinney, John W, Robertson, David L, & Lovell, Simon C. 2008. Protein-protein interaction networks and biology—what's the connection? *Nature biotechnology*, **26**(1), 69–72.

Han, Jing-Dong J, Dupuy, Denis, Bertin, Nicolas, Cusick, Michael E, & Vidal, Marc. 2005. Effect of sampling on topology predictions of protein-protein interaction networks. *Nature biotechnology*, **23**(7), 839–844.

Hayes, Wayne, Sun, Kai, & Przulj, Nataša. 2015. *On the suitability of graphlet-based measures for biological network comparison.* Unpublished.

Heath, Thomas Little, *et al.* . 1956. *The thirteen books of Euclid's Elements.* Vol. 3. Courier Corporation.

Higham, Desmond J, Rašajski, Marija, & Pržulj, Nataša. 2008. Fitting a geometric graph to a protein–protein interaction network. *Bioinformatics*, **24**(8), 1093–1099.

Ispolatov, I, Krapivsky, PL, Mazo, I, & Yuryev, A. 2005. Cliques and duplication–divergence network growth. *New journal of physics*, **7**(1), 145.

Jeong, Hawoong, Tombor, Bálint, Albert, Réka, Oltvai, Zoltan N, & Barabási, A-L. 2000. The large-scale organization of metabolic networks. *Nature*, **407**(6804), 651–654.

Jeong, Hawoong, Mason, Sean P, Barabási, A-L, & Oltvai, Zoltan N. 2001. Lethality and centrality in protein networks. *Nature*, **411**(6833), 41–42.

Joy, Maliackal Poulo, Brock, Amy, Ingber, Donald E, & Huang, Sui. 2005. High-betweenness proteins in the yeast protein interaction network. *BioMed Research International*, **2005**(2), 96–103.

Keener, James P. 1993. The Perron-Frobenius theorem and the ranking of football teams. *SIAM review*, **35**(1), 80–93.

Kendall, Maurice G, & Smith, B Babington. 1939. The problem of m rankings. *The annals of mathematical statistics*, **10**(3), 275–287.

Klemm, K., & Eguiluz, V.M. 2002. Highly clustered scale-free networks. *Physical Review E*, **65**(3), 036123.

Kolaczyk, Eric D. 2009. *Statistical analysis of network data*. Springer.

Kolaczyk, Eric D, & Krivitsky, Pavel N. 2011. On the question of effective sample size in network modeling. *arXiv preprint arXiv:1112.0840*.

Kossinets, Gueorgi. 2006. Effects of missing data in social networks. *Social networks*, **28**(3), 247–268.

Krapivsky, Paul L, & Redner, Sidney. 2001. Organization of growing random networks. *Physical Review E*, **63**(6), 066123.

Krapivsky, Paul L, Redner, Sidney, & Leyvraz, Francois. 2000. Connectivity of growing random networks. *Physical review letters*, **85**(21), 4629.

Kuchaiev, Oleksii, & Przulj, Natasa. 2009. Learning the structure of protein-protein interaction networks. *Pages 39–50 of: Pacific Symposium on Biocomputing*, vol. 14. Citeseer.

Kuchaiev, Oleksii, Stevanović, Aleksandar, Hayes, Wayne, & Pržulj, Nataša. 2011. GraphCrunch 2: Software tool for network modeling, alignment and clustering. *BMC bioinformatics*, **12**(1), 24.

Kumar, Ravi, Raghavan, Prabhakar, Rajagopalan, Sridhar, & Tomkins, Andrew. 1999. Trawling the Web for emerging cyber-communities. *Computer networks*, **31**(11), 1481–1493.

Laskar, Renu. 1969. Eigenvalues of the adjacency matrix of cubic lattice graphs. *Pacific Journal of Mathematics*, **29**(3), 623–629.

Leinhardt, Samuel. 1976. Local structure in social networks. *Sociological methodology*, **7**, 1–45.

Li, Loretta, & Schucany, William R. 1975. Some properties of a test for concordance of two groups of rankings. *Biometrika*, **62**(2), 417–423.

Li, Lun, Alderson, David, Doyle, John C, & Willinger, Walter. 2005. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, **2**(4), 431–523.

Loong, Tze-Wey. 2003. Understanding sensitivity and specificity with the right side of the brain. *BMJ: British Medical Journal*, **327**(7417), 716.

Mashaghi, AR, Ramezanpour, Abolfazl, & Karimipour, V. 2004. Investigation of a protein complex network. *The European Physical Journal B-Condensed Matter and Complex Systems*, **41**(1), 113–121.

MATLAB. 2010. *version 7.11.1 (R2010b)*. Natick, Massachusetts: The MathWorks Inc.

Melancon, Guy. 2006. Just how dense are dense graphs in the real world?: a methodological note. *Pages 1–7 of: Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*. ACM.

Middendorf, Manuel, Ziv, Etay, & Wiggins, Chris H. 2005. Inferring network mechanisms: The Drosophila melanogaster protein interaction network. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(9), 3192–3197.

Molloy, Michael, & Reed, Bruce A. 1995. A critical point for random graphs with a given degree sequence. *Random structures and algorithms*, **6**(2/3), 161–180.

Moody, James. 2004. The structure of a social science collaboration network: Disciplinary cohesion from 1963 to 1999. *American sociological review*, **69**(2), 213–238.

Nacher, Jose C, Hayashida, Morihiro, & Akutsu, Tatsuya. 2009. Emergence of scale-free distribution in protein–protein interaction networks based on random selection of interacting domain pairs. *BioSystems*, **95**(2), 155–159.

Newman, M. E. J. 2001a. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, **98**(2), 404–409.

Newman, Mark EJ. 2001b. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical review E*, **64**(1), 016132.

Newman, Mark EJ. 2002. Assortative mixing in networks. *Physical review letters*, **89**(20), 208701.

Newman, Mark EJ. 2003. Mixing patterns in networks. *Physical Review E*, **67**(2), 026126.

Newman, Mark EJ. 2004. Analysis of weighted networks. *Physical Review E*, **70**(5), 056131.

Newman, Mark EJ. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, **74**(3), 036104.

Newman, Mark EJ. 2008. The mathematics of networks. *The new palgrave encyclopedia of economics*, **2**(2008), 1–12.

Nicosia, Vincenzo, De Domenico, Manlio, & Latora, Vito. 2013. Characteristic exponents of complex networks. *arXiv preprint arXiv:1306.3808*.

Opsahl, Tore, & Panzarasa, Pietro. 2009. Clustering in weighted networks. *Social networks*, **31**(2), 155–163.

Özgür, Arzucan, Özgür, Levent, & Güngör, Tunga. 2005. Text categorization with class-based and corpus-based keyword selection. *Pages 606–615 of: Computer and Information Sciences-ISCIS 2005*. Springer.

Pach, János. 1999. *Geometric graph theory*. Tech. rept. Cambridge University Press.

Pastor-Satorras, Romualdo, Smith, Eric, & Solé, Ricard V. 2003. Evolving protein interaction networks through gene duplication. *Journal of Theoretical biology*, **222**(2), 199–210.

Penrose, M. 2003. *Random geometric graphs*. Vol. 5. Oxford University Press Oxford, UK:.

Przulj, N, Corneil, D G, & Jurisica, I. 2004. Modeling interactome: scale-free or geometric? *Bioinformatics*, **20**(18), 3508–15.

Przulj, Natasa. 2007. Biological network comparison using graphlet degree distribution. *Bioinformatics*, **23**(2), e177–83.

Pržulj, Nataša. 2010. Erratum to Biological Network Comparison Using Graphlet Degree Distribution. *Bioinformatics*, **26**(6), 853–854.

Pržulj, Nataša, & Higham, Desmond J. 2006. Modelling protein–protein interaction networks via a stickiness index. *Journal of The Royal Society Interface*, **3**(10), 711–716.

Raman, Karthik. 2010. Construction and analysis of protein-protein interaction networks. *Autom Exp*, **2**(1), 2.

Schüler, Andreas, & Bornberg-Bauer, Erich. 2011. The evolution of protein interaction networks. *Pages 273–289 of: Data Mining in Proteomics*. Springer.

Seglen, Per O. 1992. The skewness of science. *Journal of the American Society for Information Science*, **43**(9), 628–638.

Shen-Orr, Shai S, Milo, Ron, Mangan, Shmoolik, & Alon, Uri. 2002. Network motifs in the transcriptional regulation network of Escherichia coli. *Nat Genet*, **31**(1), 64–8.

Simon, Herbert A. 1955. On a class of skew distribution functions. *Biometrika*, 425–440.

Sole, Ricard V., Pastor-Satorras, Romualdo, Smith, Eric, & Kepler, Thomas B. 2002. A MODEL OF LARGE-SCALE PROTEOME EVOLUTION. *Advances in Complex Systems*, **05**(01), 43–54.

Song, Chaoming, Havlin, Shlomo, & Makse, Hernan A. 2005. Self-similarity of complex networks. *Nature*, **433**(7024), 392–395.

Stehman, Stephen V. 1997. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, **62**(1), 77–89.

Stewart, Gilbert W. 1993. On the early history of the singular value decomposition. *SIAM review*, **35**(4), 551–566.

Stomakhin, Alexey, Short, Martin B, & Bertozzi, Andrea L. 2011. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, **27**(11), 115013.

Strogatz, Steven H. 2001. Exploring complex networks. *Nature*, **410**(6825), 268–276.

Stumpf, Michael PH, Wiuf, Carsten, & May, Robert M. 2005. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(12), 4221–4224.

Su, Xianchuang, Jin, Xiaogang, Min, Yong, Mo, Linjian, & Yang, Jiangang. 2011. A Curve Shaped Description of Large Networks, with an Application to the Evaluation of Network Models. *PLoS ONE*, **6**(5), e19784.

Tanaka, Reiko, Yi, Tau-Mu, & Doyle, John. 2005. Some protein interaction data do not exhibit power law statistics. *FEBS letters*, **579**(23), 5140–5144.

Vázquez, A., Flammini, A., Maritan, A., & Vespignani, A. 2003. Modeling of Protein Interaction Networks. *Complexus*, **1**(1), 38–44.

Vogelstein, Joshua T, Roncal, William Gray, Vogelstein, R Jacob, & Priebe, Carey E. 2013. Graph classification using signal-subgraphs: Applications in statistical connectomics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(7), 1539–1551.

Von Mering, Christian, Krause, Roland, Snel, Berend, Cornell, Michael, Oliver, Stephen G, Fields, Stanley, & Bork, Peer. 2002. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, **417**(6887), 399–403.

Waldorp, Lourens J, & Schmittmann, Verena D. 2015. Computing Assortative Mixing by Degree with the-Metric in Networks Using Linear Programming. *Journal of Applied Mathematics*, **2015**.

Wang, Xiaomin, Latapy, Matthieu, & Soria, Michele. 2012. Deciding on the Type of the Degree Distribution of a Graph from Traceroute-like Measurements. *International Journal of Computer Networks & Communications*, **4**(3), 151–167.

Watts, Duncan J., & Strogatz, Steven H. 1998. Collective dynamics of /'small-world/' networks. *Nature*, **393**(6684), 440–442.

Winer, Dave. 2007. How to avoid sounding like a monkey. *Scripting News*.

Xenarios, Ioannis, Salwinski, Lukasz, Duan, Xiaoqun Joyce, Higney, Patrick, Kim, Sul-Min, & Eisenberg, David. 2002. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic acids research*, **30**(1), 303–305.

Yule, G Udny. 1926. Why do we sometimes get nonsense-correlations between Time-Series?–a study in sampling and the nature of time-series. *Journal of the royal statistical society*, 1–63.

Zhang, Bin, & Horvath, Steve. 2005. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, **4**(1).

Zhu, Han, Wang, Xinran, & Zhu, Jian-Yang. 2003. Effect of aging on network structure. *Physical Review E*, **68**(5), 056121.