UNIVERSITY OF CALIFORNIA,
IRVINE

Graphlet Analysis Of Networks

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Sridevi Kamla Maharaj

Dissertation Committee:
Professor Wayne Hayes, Chair
Professor Sandy Irani
Professor Zeba Wunderlich

2018

© 2018 Sridevi Kamla Maharaj

DEDICATION

Amidst the darkness, the light that glimmers faintly and then shines strongly and brightly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

My heartfelt thank you to my advisor Professor Wayne Hayes for his colossal support and guidance over the past few years, for instilling confidence in me, helping me rediscover love for learning, and showing me a fruitful path to follow. Thank you for creating a comfortable, collaborative, informal work atmosphere.

Thank you to the rest of my dissertation committee - Professor Sandy Irani and Professor Zeba Wunderlich for your feedback and comments.

Thank you to my colleague and office-mate for the past year, Pedro Silva for making an otherwise cold office into a much warmer work environment.

Thank you to all the undergraduate students from Wayne's research group who made my mentoring experience both exciting and educational.

Thank you to the CS Department and International Center at UCI.

CURRICULUM VITAE

Sridevi Kamla Maharaj

EDUCATION

Doctor of Philosophy in Computer Science — **expected 2018**
University of California, Irvine — *Irvine, CA*

Master of Science in Computer Science — **2014**
University of California, Irvine — *Irvine, CA*

Bachelor of Science in Mathematics and Computer Science — **2011**
University of Miami — *Coral Gables, FL*

RESEARCH EXPERIENCE

Graduate Research Assistant — **Summer 2018**
University of California, Irvine — *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant — **2012–2018**
University of California, Irvine — *Irvine, CA*

REFEREED JOURNAL PUBLICATIONS

Comparing Different Graphlet Measures for Evaluating Network Model Fits to BioGRID PPI Networks **2018**
(submitted)

REFEREED CONFERENCE PUBLICATIONS

Firing Squad Synchronization Problem - Partial Solutions on Rings **2018**
(submitted)

BLANT - Sampling Graphlets in a Flash **June 2018**
q-bio

REFEREED POSTERS

Modeling the Structure of BioGRID PPI Networks **December 2018**
Rocky Mountain Bioinformatics Conference

BLANT - Sampling Graphlets in a Flash **June 2018**
q-bio

ABSTRACT OF THE DISSERTATION

Graphlet Analysis Of Networks

By

Sridevi Kamla Maharaj

Doctor of Philosophy in Computer Science

University of California, Irvine, 2018

Professor Wayne Hayes, Chair

Over the past decade, the study of graphlets has emerged as a useful tool in the study of networks. Graphlets are small, induced, connected subgraphs on a larger graph. They aid in quantifying the structure of networks, identifying functionality of sub-regions within a network, and understanding how relationships and interactions represented by the network may evolve over time. In this study, we approach a variety of problems from the vantage point of graphlets. First, we model the structure of BioGRID protein-protein interaction (PPI) networks using different network models and assess their fitness primarily via their graphlet topology. We also briefly explore the relationships between various network comparison measures and find there is little agreement between them. Despite this disagreement, we find that the models most suited to modeling PPI networks have changed as the data evolve and that it is the STICKY and scale-free based models which best fit current PPI networks. Secondly, as the PPIs (and other networks) are constantly updated, they become larger and denser. As network data is growing in volume (some networks have on the order of hundreds of thousands of nodes and millions of edges), exhaustive enumeration of graphlets becomes infeasible and hence, in order to perform graphlet analysis on networks, we must sample graphlets from networks. We explore four different graphlet sampling techniques. We highlight the advantages and disadvantages of different sampling approaches and show that a Markov Chain Monte Carlo approach to graphlet sampling is best for approximating

the proportion of each graphlet within a network. In addition, we find that though they have different biases, all four techniques are able sample a sufficiently diverse set of graphlets that graphlet-based network comparison measures are still able to distinguish between different network types. From sampling, we are also able to distinguish graphs of much higher density, which would ordinarily take weeks or months of CPU time to fully enumerate. Thirdly, we demonstrate a need for the development of synthetic network generators by showing that that many state-of-the-art synthetic network generators do not consistently reproduce networks matching traditional properties of real-world networks and graphlet distributions of the real-world networks. Finally, we analyze the brain fMRI connectivity in children with OCD and a control (CON) group. We find that the network connectivity is completely different in both networks and we highlight some of their differences, including hub nodes, different correlations between connectivity paths and physical distances in the brain, the occurrence of cliques and circuits and graphlet distributions. We also model the OCD and CON connectivity networks using theoretical models and find that the CON network is fit best by a Geometric with Gene Duplication and Mutation network while the OCD network has no best fit.

Chapter 1

Introduction

1.1 Networks and Notation

A network or graph $G(n, m)$ is a set of n nodes (vertices) and m edges connecting the nodes. Edges can be directed or undirected, and may be weighted or unweighted. In this study, we focus on biological networks and these are generally simplified to be undirected and unweighted. The notation used to describe networks is shown in Table 1.1.

Notation	Definition
$G(n, m)$	A network G with n nodes and m edges
$G(V, E)$	A network G with vertex set V and edge set E
$Adj(G)$	The adjacency matrix representation of G
$Deg(v)$	The degree of a node v
$Neigh(v)$	The set of neighbors of a node v
(u, v)	The edge whose endpoints are u and v

Table 1.1: Notation used for graphs.

Networks have been used extensively to represent many kinds of real-world phenomena such as transportation routes [Bell and Iida, 1997], trade [Furusawa and Konishi, 2007, **?**], social interactions and relationships [Kross et al., 2013, Zywica and Danowski, 2008], academic

collaborations [Newman, 2001], citations Rice et al. [1988], and biological processes and connectomes [Luck et al., 2017, Karlebach and Shamir, 2008, Milano et al., 2017]. The network representation of a process or system can aid in formalizing and precisely defining the system. Analyzing and measuring the network topology – patterns and regularities of connections between nodes – have been shown to aid in answering behavioural science problems [Kross et al., 2013, Zywica and Danowski, 2008], understanding the overall function, structure and behaviour of biological systems [Li et al., 2013, Janjić et al., 2014, Sporns, 2010], observing patterns of trade and commodity costs [Yaveroğlu et al., 2014] and understanding how loss of biodiversity and species invasions may affect ecosystems [Bornholdt and Schuster, 2006]. Although some topological measures hint at being able to retrieve specific functional information, none appear to give robust enough results to be of general use, with perhaps the possible exception of *graphlets* [Pržulj et al., 2004, Kuchaiev et al., 2010, Davis et al., 2015]. In this thesis, we focus on graphlet analysis of various networks.

1.2 Graphlets

Definition 1. *Given a graph $G(n, m)$ on n nodes and m edges, a* ***k-graphlet*** *is an induced, subgraph g on any set of k connected nodes from G, where k is typically between $2-5$ [Pržulj et al., 2004].*

Definition 2. *An* ***automorphism*** *on a k-graphlet (or graph) g is a bijective function $f : \{1..k\} \to \{1..k\}$ such that (u, v) is an edge in g if and only if $(f(u), f(v))$ is an edge in the image of g under f.*

Definition 3. *An* ***orbit*** *of a node u of g is defined as $O_u = \{v \in V_g | v = \pi(u)$, for each automorphism π of $g\}\}$. In other words, those nodes of a graphlet which when swapped result in an autmorphism of the same graphlet, are said to belong to the same orbit.*

Figure 1.1 shows all the graphlets on 2, 3, 4 and, 5 nodes including their automorphism orbits [Pržulj, 2007]. Automorphisms enumerate all the different ways of drawing the same graph. For example, we can draw G_{15} as it is drawn in Figure 1.1 or as a star but both drawings are cycles of length 5.

A more systematic way of ordering the k-graphlets was used in Melckenbeeck et al. [2017a]. Each k-graphlet g can be represented as an adjacency matrix. We form the binary representation of g by reading the lower triangular matrix of $Adj(g)$. Due to automorphisms on g, there may be more than one adjacency matrix to represent g. The lowest binary representation is called the *canonical representation* and is used for identification of any automorphsim of g, as shown in Figure 1.2. The same ordering scheme was used in Hasan et al. [2017] but extended to *graphettes* (i.e. graphlets which may be connected or disconnected). Using this system, connected k-graphlets can be ordered according to the magnitude of the binary representation of their canonical forms. This ordering is shown in Figure 1.1 for $3, 4$, and 5-graphlets and is employed by Melckenbeeck et al. [2017a], Hasan et al. [2017]. Figure 1.3 shows the number of canonicals graphettes and orbits for values of k up to 12 as well as the storage requirements for listing them using binary representation.

In order to automatically enumerate the orbits of each graphette (or graphlet), Hasan et al. [2017] generated all automorphisms of graphlet g, split each automorphism into its cycles, and merged the cycles from different automorphisms to form orbits. A lengthy proof is presented in Hasan et al. [2017] to show that (i) each node of g is part of a cycle and (ii) each node is part of exactly one orbit. We present a novel, much shorter proof, noting that, the set of automorphisms on g is the automorphism group *Aut(g)* with composition as its cation and the equivalence classes of each k-graphlet g under action from $Aut(g)$ are the **orbits** of g. The propositions in Hasan et al. [2017] are properties of a group structure:

Proposition 1. *For each node $u \in V(g)$, and each automorphism $\pi : V(g) \to V(g)$, there exists a $\lambda > 0$ such that $\pi^{\lambda}(u) = u$.*

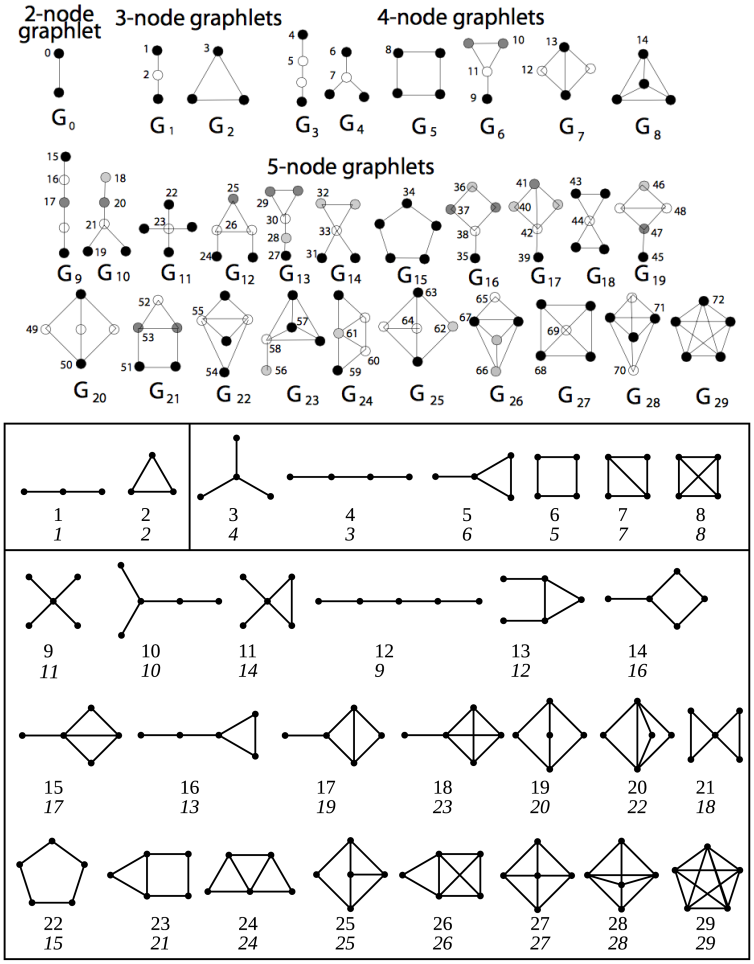


Figure 1.1: **Top**: All 2, 3, 4, and 5-node graphlets. The numbers in normal font represent the graphlet ordering of Pržulj et al. [2004]. Within each graphlet, *automorphism orbits*—nodes that are topologically identical within the graphlet—have the same shade. There are 73 distinct orbits, numbered 0 through 72, identified by numbers next to one of the nodes of each orbit. **Bottom**: All 3, 4, and 5-node graphlets according to lower triangular adjacency matrix canonical ordering [Melckenbeeck et al., 2016]. They are the same as the $3, 4$, and 5-graphlets shown at the top but ordered more systematically. The numbers to the bottom of each graphlet in regular font represents the graphlet ID according to the ordering used in Melckenbeeck et al. [2017a], Hasan et al. [2017] while the numbers to the bottom of each graphlet in italics represent the graphlet ID according to the ordering of Pržulj et al. [2004].

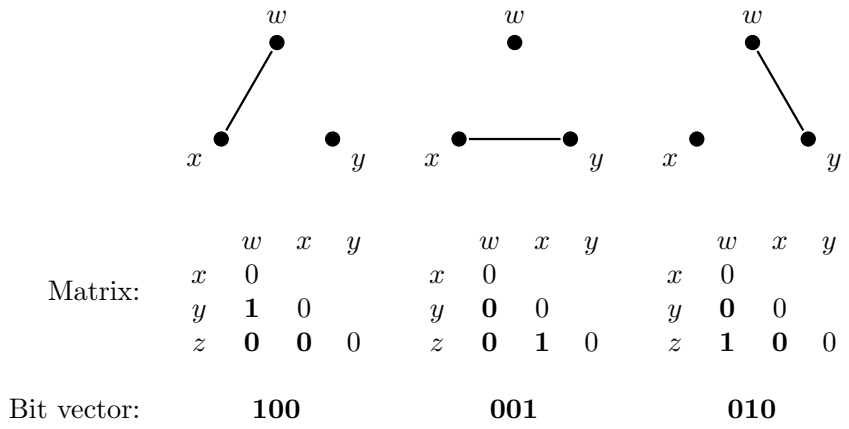


Figure 1.2: The three graphettes shown here are automorphisms for each other but their binary representations are completely different. Since the middle representation is the smallest (001), that form of the graphette is chosen as the canonical representing the group.

k	bits *b*(*k*)	#Graphs $2^{b(k)}$	Space $b(k)2^{b(k)}$	#Canonicals *NC*(*k*)	#Orbits
1	0	1	0	1	1
2	1	2	0.25 B	2	2
3	3	8	3 B	4	6
4	6	64	48 B	11	20
5	10	1 K	1.25 KB	34	90
6	15	32 K	60 KB	156	544
7	21	2 M	5.25 MB	1044	5096
8	**28**	**256 M**	**896 MB**	**12346**	**79264**
9	36	64 G	288 GB	274668	2208612
10	45	32 T	180 TB	12005168	113743760
11	55	32 P	220 PB	1018997864	10926227136
12	66	64 E	528 EB	165091172592	1956363435360

https://doi.org/10.1371/journal.pone.0181570.t002

Figure 1.3: For each value of k: The number of bits $b(k) = \frac{k(k-1}{2}$ required to store the lower-triangle of the adjacency matrix for an undirected k-graphette; the number of such k-graphettes counting all isomorphisms which is just 2b(k); the number of canonical k-graphettes (this will be the number of unique entries in the above lookup table [Sloane, a], and up to k = 8, 14 bits is sufficient); and the total number of unique automorphism orbits (up to k = 8, 17 bits is sufficient) [Sloane, b]. Note that up to k = 8, together the lookup table for canonical graphettes and their canonical orbits fits into 31 bits, allowing storage as a single 4-byte integer, with 1 bit to store whether the graphette is connected (i.e., also a graphlet). The suffixes K, M, G, T, P, and E represent exactly 210, 220, 230, 240, 250 and 260, respectively. https://doi.org/10.1371/journal.pone.0181570.t002

Proof. Since G is a finite graph, all elements of $Aut(G)$ have finite order. The result follows. □

Proposition 2. *The orbits are disjoint. (In other words, each node appears in exactly one orbit.)*

Proof. Any automorphism $\pi \in Aut(g)$ on g partitions the nodes into disjoint sets and these sets are *cycles*. Furthermore, O_u, the orbit of node u is a collection of the cycles of u under group action. □

Definition 4. *The **graphlet-orbit signature** of a node v is a vector of the number of each graphlet orbit to which v belongs [Milenković and Pržulj, 2008].*

Definition 5. *The **graphlet orbit signature matrix**, S_G is a matrix in which each row r of S_G represents the graphlet orbit signature of node r in G. The dimensions of S_G are $n \times t$, where n is the number of nodes in G and t is the number of orbits. The graphlet topology of a network can be approximated by its graphlet orbit signature matrix.*

Graphlets have become very popular in the last decade as a way of quantifying local structure within networks, especially biological networks. Graphlets and their orbit signatures have been used to: (i) aid global alignments [Kuchaiev et al., 2010, Kuchaiev and Pržulj, 2011b, Milenković et al., 2010, Saraph and Milenković, 2014, Vijayan et al., 2015, Malod-Dognin and Pržulj, 2015, Mamano and Hayes, 2017], (ii) perform alignment-free comparison of networks [Kashtan et al., 2004, Pržulj et al., 2004, Pržulj and Hayes, 2006, Pržulj, 2007, Pržulj and Milenković, 2009, Hayes et al., 2013, Yaveroğlu et al., 2015], (iii) as a heuristic in the analysis of brain connectomes [Sporns and Kötter, 2004, Sporns, 2012], and (iv) to recover both functional and phylogenetic information [Kuchaiev et al., 2010, Davis et al., 2015]. Many tools have also been developed to enumerate and count all the 2, 3, 4 and 5-graphlets in a network and output its grahplet orbit signature matrix Hočevar and Demšar [2014],

Melckenbeeck et al. [2016, 2017b], Hočevar and Demšar [2017]. The time complexity for counting all k-graphlets for $k = 3, 4$, and 5 is:

$$O(nd^{k-1}), \text{ where d is the maximum degree in G } [\textit{Shervashidzeet al.}, \text{ 2009}]. \tag{1.1}$$

1.2.1 Network Comparison Measures using Graphlets

It has long been known that a complete network comparison between large networks is computationally intractable, as it is an NP-complete problem [Cook, 1971]. Thus, many heuristics have been brought forward to approximate the similarities and differences. Two ways to compare networks are: (i) Alignment-Based Network Comparison (ii) Alignment-Free Network Comparison. In the first approach, algorithms are designed to find a map between a non-trivial subset of nodes from two (or more) networks, while maximizing the edge relationships which are preserved by the mapping. This approach is useful in identifying, for example, regions of a biological network that remain invariant throughout evolution [Milenković et al., 2013, Ibragimov et al., 2014, 2013, Kelley et al., 2003, Kuchaiev and Pržulj, 2011a, Liao et al., 2009, Neyshabur et al., 2013, Saraph and Milenković, 2014, Vijayan et al., 2015, Kuchaiev et al., 2010, Malod-Dognin and Pržulj, 2015, Zhang and Skolnick, 2005, Malod-Dognin et al., 2017, Mamano and Hayes, 2017]. On the other hand, alignment-free network comparison aims to assess the topological similarity of networks as a whole, regardless of edge relationships and node-to-node mappings. This approach is useful in evaluating the fit of a random network model to a type of real-world network, such as a biological network [Pržulj and Higham, 2006, Pržulj, 2007, Hayes et al., 2013], tracking the dynamics of a time-series network [Garlaschelli and Loffredo, 2005, Yaveroğlu et al., 2014], clustering networks based on their topological similarities [Milo et al., 2004, Yaveroğlu et al., 2014] and reconstructing phylogenetic relationships of various organisms based on network similarity [Ali et al., 2014].

In alignment-free methods, computations are performed to obtain a score representing the overall similarity (or difference) between the networks being considered. Early work in alignment-free network comparison used global network properties such as degree-distribution, clustering coefficient, diameter, etc [Watts and Strogatz, 1998a, Newman, 2003, Khanin and Wit, 2006, Cabrera-Vera et al., 2003, Emmert-Streib et al., 2016]. These turned out to be grossly ineffective at separating different types of networks [Pržulj, 2007, Yaveroğlu et al., 2014]. Many distance measures and similarity measures have been proposed which utilize various local network properties as heuristics, including graph spectra [Wilson and Zhu, 2008, Thorne and Stumpf, 2012], topological indices [Dehmer et al., 2014], motifs [Milo et al., 2002], graphlets [Pržulj et al., 2004, Pržulj, 2007] and ego-networks [Ali et al., 2014]. Graphlets and in particular, graphlet-orbit signatures, have become a popular and effective heuristic [Hayes et al., 2013, Faisal et al., 2015] and have been used both in alignment [Milenković et al., 2013, Milenković and Pržulj, 2008] and non-alignment comparison methods [Pržulj et al., 2004], as well as to classify different types of networks [Yaveroğlu et al., 2014].

In order to evaluate the similarity between networks, several distance and similarity measures based on graphlet counts, graphlet distribution and graphlet orbit signatures have been developed and used to quantify local topology within different kinds of networks, including biological networks Pržulj et al. [2004], Pržulj [2007], Przulj [2010], Shervashidze et al. [2009], Yaveroğlu et al. [2014]. Graphlet-based network comparison measures can be split into two classes: distance measures, where smaller distances mean higher similarity, and similarity measures (or agreements), where larger scores mean higher similarity. Relative Graphlet Frequency Distance (RGFD) compares log counts of $2, 3, 4$ and, 5-graphlets within two networks [Pržulj et al., 2004]. Graphlet Degree Distribution Agreement (GDDA) compares the similarity between distributions of each orbit within two networks and ranges between 0 and 1 [Pržulj, 2007]. Graphlet Kernel (GK) is the dot product of normalized graphlet counts and hence ranges between 0 and 1 [Shervashidze et al., 2009]. Graphlet Correlation Distance (GCD) is the Euclidean distance between correlations of graphlet-orbit signature

vectors [Yaveroğlu et al., 2014]. There were 11 orbits which were claimed to be sufficient for distinguishing between networks types Yaveroğlu et al. [2014]. As such, a GCD score can range between 0 and $2\sqrt{73}$ if all 73 orbits of $2, 3, 4,$ and 5-graphlets are used, or between 0 and $2\sqrt{11}$ if the 11 orbits are used instead. Complete formulae for RGFD, GCD, GDDA and, GK are shown in Table 1.2.

Measure	**Input Variables**	**Formula**
Relative Graphlet Frequency Distance (RGFD)	Graphlet Counts $\mathbf{u}$, $\mathbf{v}$	$\sum_i \mid -log(\frac{u_i}{\sum_j u_j}) + log(\frac{v_i}{\sum_j v_j})\mid$
Graphlet Correlation Distance (GCD)	Correlation Matrices of Signatures, C_1, C_2	$\sum_{i,j}(C_1[i,j] - C_2[i,j])^2$
Graphlet Kernel (GK)	Graphlet Counts $\mathbf{u}$, $\mathbf{v}$ or Unit Vectors of the Graphlet Counts	$\frac{u \cdot v}{\|\|u\|\|.\|\|v\|\|}$
Graphlet Degree Distribution AGreement (GDDA)	Graphlet orbit degree distributions D^j_G, D^j_H $S^j_G(k) = \frac{D^j_G(k)}{k}$ $N^j_G(k) = \frac{S^j_G(k)}{\sum_k S^j_G(k)}$	$1 - \sqrt{(\sum_{k=0}^{\infty}(N^j_G(k) - N^j_H(k))^2)}$
Senatorial Graphlet Kernel (SGK)	Graphlet Counts Graphlet Counts $\mathbf{u}$, $\mathbf{v}$ or Unit Vectors of the Graphlet Counts	$\sum_i \frac{min(u_i,v_i)}{max(u_i,v_i)}$

Table 1.2: The formulae for computing RGFD, GCD, GK, GDDA and SGK are given above. The first two measures are distance measures and hence, a lower score is preferable between two networks of similar graphlet structure. The other three measures are similarity measures ranging between 0 and 1, from different to more similar. Consider networks G and H. For RGFD, the total number of each type of graphlet for each network is counted and stored in vectors u and v. These vectors are used in the computation. For GCD, correlations are taken pairwise across each column of the graphlet orbit signature matrix to form a correlation matrix. Computing GDDA is more involved. It begins with finding the distribution of orbits of type j in each graph $D^j_G(k)$: count the number of nodes which belong to orbits of type j exactly k times, for $k \in \mathbb{Z}_{\geq 0}$ and dividing each $D^j_G(k)$ by k, then normalization and then finally, the computation. For GK and SGK, we use the same vectors as for RGFD or we can convert u and v into unit vectors.

1.3 Outline and Summary of Contribution

In Chapter 2, we introduce a new graphlet-based alignment-free network comparison measure which attempts to give equal importance to all graphlets. We show that if it assesses two networks as 'asymptotically' the same, then so will the Graphlet Kernel measure. We begin chapter 3 with a presentation of different types of theoretical network models and discuss a few inconsistencies in the previous literature when modeling protein-protein interaction networks. Following this, we endeavour to re-model the latest protein-protein

interaction networks obtained from BioGRID [Chatr-Aryamontri et al., 2017] using several theoretical models that have been introduced in the literature. We assess their fits using several graphlet-based network measures and compare the agreement and performance of the measures themselves. Chapter 4 begins with a brief discussion on the infeasibility of exhaustive graphlet and orbit enumeration in this age of growing data. We then survey graphlet sampling algorithms. We examine and compare the extent to which four different graphlet sampling techniques are able to replicate the graphlet distribution of the original network. We also test whether sampled graphlets (even with a bias) from different methods are sufficient for distinguishing one network type from another. In chapter 5, we demonstrate that there is a need for synthetic network generators to be developed as the current state-of-the-art network generators reproduce synthetic networks which have significantly different properties from corresponding real-world networks. In chapter 6, we study the brain connectivity patterns in children patients with Obsessive Compulsive Disorder (OCD). We overview the need for deeper understanding of brain connectivity in patients with OCD and note that previous studies on OCD patients used basic graph theoretic analysis. We seek to analyze the connections using more advanced graph analysis tools, including graphlets. We also introduce the k-Graphlet Edge Hamming Distance, prove some of its basic properties and utilize it in the analysis of the OCD network. We note significant differences of various kinds in the OCD connectivity network when compared with connectivity of healthy children patients. Finally, in chapter 7, we briefly discuss the future direction of the work presented in the previous chapters.

1.4 Real World Network Data

The real-world data used throughout this thesis is described below.

1.4.1 Biological Networks

Several types of biological networks are used and referred to:

- 1 Collin's Yeast Protein-Protein Interaction Network with 1004 nodes and 8323 edges [Collins et al., 2007].
- 9 Protein-Protein Interaction Networks from BioGRID described later in Table 3.1. [Chatr-Aryamontri et al., 2017].
- 2 Brain fMRI Networks: OCD and Control from USC Multimodal Connectivity Database described in Table 1.3 [Brown et al., 2012].
- 620 Brain Networks from the Network Repository ranging from 29 to $976K$ nodes and 44 to $268M$ edges [Rossi and Ahmed, 2015a].
- 535 Gene-μRNA networks from Tokar et al. [2017].
- 600 Chemoinformatics networks from Rossi and Ahmed [2015a].
- 78 Foodweb networks reprsenting predator-prey relationships between various organisms from The Index of Complex Networks [Clauset et al., 2016].

Network	Number of Nodes	Number of Edges	Density
OCD_19_Mean (OCD)	200	1990	0.1
CON_17_Mean (CON)	200	1990	0.1

Table 1.3: The two networks described here are from the USC Multimodal Connectivity Database representing fMRIs from the brains of 19 child patients with Obsessive Compulsive Disorder (OCD) and 17 children without OCD, ie the Control (CON) group.

1.4.2 Other Types of Networks

A wide array of networks other than biological networks was also used:

- 733 Autonomous Systems networks taken from the Stanford SNAP database representing routers of the Internet which communicate with each other based on the Border Gateway Protocal logs [Leskovec and Sosič, 2016] .

- 271 Combinatorial problem networks from The University of Florida Sparse Matrix Collection [Davis and Hu, 2011].

- 98 Facebook networks from the Network Repository representing social relationships between people at various university campuses [Rossi and Ahmed, 2015a].

- 34 Retweet networks from The Network Repository [Rossi and Ahmed, 2015a].

- 19 Technology networks representing various tools and their library dependencies The Index of Complex Networks [Clauset et al., 2016].

Chapter 2

A New Graphlet-Based Network Comparison Measure

2.1 Senatorial Graphlet Kernel

The Graphlet Kernel (GK) (cf Table 1.2) can be dominated by large members: clearly, since the numerator is a dot product, if two graphs have the same graphlet g grossly dominating the counts among all graphlets, then the dot product will be dominated by the count of that graphlet in GK. This means that graphlets that are uncommon will have little influence on the value of the dot product, and therefore their relative difference can be large without detection from GK. While one could argue that the most frequent graphlets are somehow the most important, one could also argue that the "minority count" graphlets are having their "voice" suppressed by the majority graphlets. A good analogy is with democratic governmental bodies: in both Republics, and Parliamentary democracies, there is a group of politicians who are apportioned by population (the lower house, called a "house of representatives", or the "house of parliament", respectively), and a second group called the "upper house"

or Senate, which is apportioned by locale, independent of the population of the locale. So for example in the US, the State of California has a population of about 30 million, while Delaware has a population of about 1 million; accordingly, California has 55 representatives in the lower house and Delaware has only 3. However both states (and all other states) have exactly 2 Senators, regardless of population.

The existing GK is like a lower house: the most populous graphlets dominate the measure. While this may be appropriate in some cases, it is certainly not the only way to share and weight votes. To that end, we introduce the *Senatorial Graphlet Kernel*, in which each type of graphlet has an equal say in measuring the similarity between two graphs, regardless of how frequently that graphlet occurs. SGK is designed to be sensitive to small differences in graphlet proportions. Given two graphs G_1, G_2, with unit vector graphlet frequencies u, v with l entries, let $r_i = \min(u_i, v_i)/\max(u_i, v_i)$. Then

$$SGK(G_1, G_2) = \frac{1}{l}\sum_{i=1}^{l} r_i \tag{2.1}$$

Note that $SGK(G_1, G_2)$ "close to 1" implies $GK(G_1, G_2)$ is also "close to 1": the SGK demands that *all* elements have similar counts, whereas GK only demands the most frequent elements to have similar counts. In other words, GK can approach 1 if counts of frequently appearing graphlets agree while infrequent ones do not. The Senatorial Graphlet Kernel is a more sensitive measure of difference than the original GK, in the sense that it can detect differences that the GK cannot, while simultaneously forcing the GK to 1 whenever the SGK is 'close to' 1. This statement is formalized in the following Proposition.

Proposition 3. *Suppose u and v are unit vectors of k-graphlet counts for graphs G_1 and G_2 respectively and l is the number of dimensions in u (both u and v must have the same length). Define $r_i = \frac{min(u_i,v_i)}{max(u_i,v_i)}, \forall i$. For any $\epsilon > 0$, suppose $\exists \delta > 0$ such that $||SGK(G_1, G_2) - 1|| < \epsilon$ and $\forall i, ||r_i - 1|| < \delta$, then $||GK(G_1, G_2) - 1|| < \epsilon^2$.*

Proof. For any $\epsilon > 0$, define $\epsilon_0 = \sqrt{\epsilon}$. Suppose $\exists \delta_0 > 0$ such that $\forall i, ||1 - r_i|| < \delta_0$ and $||SGK(G_1, G_2) - 1|| < \epsilon_0$. Writing out the terms, this means

$$\forall i, |1 - \frac{min(u_i, v_i)}{max(u_i, v_i)}| < \delta_0, \tag{2.2}$$

and

$$|\frac{1}{l}\sum(1 - \frac{min(u_i, v_i)}{max(u_i, v_i)})| < \epsilon_0. \tag{2.3}$$

We can guarantee that $\delta_0 \leq \epsilon_0$ because inequalities 2.2 and 2.3 together imply, $\frac{1}{l}|\sum(1 - \frac{min(u_i,v_i)}{max(u_i,v_i)})| < \epsilon_0, \delta_0$. From the inequality 2.2, we have

$$\forall i, 0 \leq max(u_i, v_i) - min(u_i, v_i) < \delta_0 max(u_i, v_i),$$

$$0 \leq (max(u_i, v_i) - min(u_i, v_i))^2 < \delta_0^2 max(u_i, v_i)^2,$$

Summing over all i, we get

$$0 \leq \sum_i (max(u_i, v_i) - min(u_i, v_i))^2 < \delta_0^2 \sum_i max(u_i, v_i)^2$$

Expanding the square, we get $0 \leq max(u_i, v_i)^2 + min(u_i, v_i)^2 - 2\sum_i max(u_i, v_i) min(u_i, v_i) < \delta_0^2 \sum_i max(u_i, v_i)^2$. Since u and v are unit vectors, it means $||u||, ||v|| = 1$. Hence,

$$0 \leq 2(1 - \sum_i max(u_i, v_i) min(u_i, v_i)) < 2\delta_0^2,$$

$$0 \leq 1 - \sum_i max(u_i, v_i) min(u_i, v_i) < \delta_0^2 \leq \epsilon_0^2 = \epsilon.$$

□

Alternatively, we can define $SGK(G_1, G_2)$ to be the geometric mean of the r_i's, i.e.

$$SGK(G_1, G_2) = \sqrt[l]{\prod}_{i=1}^{l} r_i,$$

which may be more suitable than the above arithmetic mean version, depending on the application. The above limit is also true for the geometric mean definition of $SGK(G_1, G_2)$ and is formalized below:

Corrollary 1. *Suppose u and v are unit vectors of k-graphlet counts for graphs G_1 and G_2 respectively and l is the number of dimensions in u (both u and v must have the same length). Define $r_i = \frac{min(u_i, v_i)}{max(u_i, v_i)}, \forall i$. For any $\epsilon > 0$, suppose $\exists \delta > 0$ such that $|| \prod i = 1^l r_i - 1|| < \epsilon$ and $\forall i, ||r_i - 1|| < \delta$, then $||GK(G_1, G_2) - 1|| < \epsilon^2$.*

Proof. We will denote the geometric mean $\sqrt[l]{\prod_{i=1}^{l} r_i}$ as GM and the arithmetic mean $\frac{1}{n}\sum_{i=1}^{l} r_i$ as AM. Since $u_i, v_i \geq 0 \forall i$, it means $GM \leq AM < 1$ and hence, if $||GM - 1|| < \epsilon$, then $||AM - 1|| < \epsilon$. By the above lemma, this implies $||GK(G_1, G_2) - 1|| < \epsilon^2$. □

2.1.1 Alternative Forms of SGK

We can also define Senatorial Graph Kernel Distance (SGKD), i.e. the distance version of SGK:

$$SGKD(G_1, G_2) = 1 - \frac{1}{l}\sum_{i=1}^{l} r_i, \tag{2.4}$$

and the Unrestricted Senatorial Graph Kernal (USGK) as follows:

$$USGK(G_1, G_2) = \frac{1}{l}\sum_{i=1}^{l} \frac{u_i}{v_i}. \tag{2.5}$$

One disadvantage of SGK in Equation 2.1 and SGKD in Equation 2.4 is that neither of these functions are monotonic on their input variables. We generally expect a good network comparison measure to be a monotonic function as it is a measure of distance (or similarity) between two networks.

Chapter 3

Evaluating Network Model Fits to BioGRID PPI Networks

3.1 Synthetic Networks

A synthetic network is a graph built algorithmically, usually including a stochastic element, and which may or may not have any relationship to a real-world network. The following definitions list several popular synthetic network models.

3.1.1 Erdös-Rényi Graphs (ER)

An ER graph $G(n, m)$ with associated probability p, is a graph in which the probability of an edge between any two nodes selected uniformly at random is p [Erdös and Rényi, 1959]. In other words, two nodes are adjacent to each other in G with probability p. The expected value of $m, E[m] = \binom{n}{2}p$. The probability p is also the expected density of G.

3.1.2 Erdös-Rényi with Degree Distribution Graphs (ERDD)

An ERDD graph $G(n, m)$ with associated degree function $f(d)$ is a graph in which the number of nodes of degree d is $f(d)$. This means $m = \frac{\sum_d f(d)}{2}$ [Milenković et al., 2008]. This means that $E[m]$ is still $\binom{n}{2}p$ (p is density), but the degree distribution is constrained to $f(d)$. In other words, the edges are no longer placed uniformly at random between nodes but instead are preferentially placed between nodes in order to create a given degree distribution across nodes.

3.1.3 Geometric Graphs (GEO)

A GEO graph $G(n, m)$ with associated radius r and dimension d, is a graph in which the n nodes are points distributed uniformly at random within a hyber-cube of volume 1 in $\mathbb{R}^d$, where d is typically 2 or 3. Two nodes are connected by an edge if they are within a radius r of each other [Penrose, 2003].

3.1.4 Geometric Gene Duplication and Mutation (GEOGD)

A GEOGD graph $G(n, m)$ has associated radius r, similar to GEO graphs. Beginning from a seed network placed in a hyper-cube of volume 1 in $\mathbb{R}^d$ (d is typically 2 or 3), nodes are added iteratively to the network by choosing a parent node uniformly at random from among the existing nodes, and placing a child node at a distance far away from the parent, in a direction chosen uniformly at random [Pržulj et al., 2010a]. The child node is then connected to the parent node and to any other node within radius r. This process stops when the the network contains n nodes. This encapsulates the idea that some children-nodes will be very similar to the parent-node and some children-nodes may gain new interactions with other nodes. There are two ways in which GEOGD networks can be generated: (i) The Expansion

Model (ii) Probability-Cutoff Model. In the Expansion model, the child node is placed at a distance at most $2r$ away from the parent node Pržulj et al. [2010a]. In the Probability-Cutoff Model, the child node is placed within the r radius of the parent with probability p but with probability $1-p$, it is placed at a distance of up to $10r$ [Pržulj et al., 2010a]. The $1-p$ probability event gives rise to a huge mutation. GEOGD graphs were invented by Pržulj et al. [2010a] with the purpose of modeling biological networks and in particular, protein-protein interaction networks, by attempting to model the concept of gene duplication and mutation are into the GEO model by allowing the possibility of an edge being shared between two nodes which are more than radius r from each other.

3.1.5 Scale Free Graphs (SF)

An SF graph $G(n, m)$ is a graph whose degree distribution asymptotically follows a power-law, ie the fraction of nodes $P(k)$ which have k connections to other nodes has the relationship $P(k)\ k^{-\gamma}$. There are various methods to generate scale-free graphs which all give slightly different structures. One popular scale-free generator is the Barabaási-Albert method, which requires two additional parameters: the number of nodes of nodes n_0 in the initial seed graph, and the number of attachments m_0 ($m_0 < n_0$) to be made at each stage of building the graph. In the Barabaási-Albert method, the graph begins with an initial, connected seed graph containing n_0 nodes. The remaining $n - n_0$ nodes are added one at a time. Each new node is connected to m_0 existing nodes with probability p proportional to the degree of the existing nodes. More formally, if the graph currently has nodes $v_1, v_2, ..., v_t$, then the probability that a new node u is connected to an existing node v_i is given by $p = \frac{deg(v_i)}{\sum_j deg(v_j)}$. In this manner, new nodes tend to attach themselves to nodes which are already heavily linked [Barabási and Albert, 1999].

3.1.6 Scale Free Gene Duplication and Divergence Graphs (SFGD)

An SFGD graph $G(n, m)$ has associated attachment probability p and detachment probability q. We begin with an initial seed graph of n_0 ($n_0 < n$) nodes. The seed graph could be just an edge. The remaining $n - n_0$ nodes are added iteratively. At each step of growing the graph, an existing node u is selected uniformly at random and there is a duplication phase followed by a divergence phase. During duplication, a new node v is added to the graph and becomes a neighbor of all the neighbors of u. The edge (u, v) is added to the graph with probability p. During the divergence phase, for any node w which is a neighbor of both u and v, we choose uniformly at random the edge (u, w) or (v, w) and remove it with probability q. This process is repeated until we have n nodes in the graph. The concept of SFGD graphs was introduced in Vázquez et al. [2003] in order to model biological networks and specifically, protein-protein interaction networks. The duplication phase captures the idea that a protein may act like a duplicate of another protein and simultaneously, may interact with its duplicate. The divergence phase attempts to model mutations in genes which will gradually produce differences in proteins, thereby altering their interactions.

3.1.7 Small World Graphs (SW)

A SW graph $G(n, m)$ is a graph in which most nodes are not adjacent to each other but the neighbors of any given node are likely to be neighbors of each other. An additional property of small world graphs is that most nodes of G can be reached from any node by taking a small number of steps. A popular generator of Small World Graphs called Watts-Strogatz Watts and Strogatz [1998b] uses additional parameters: (i) k the number of nodes that each node will connect to in a ring topology, and (ii) p The probability of re-wiring each edge. The graph first begins with a ring over n nodes. Each node is then connected with its k nearest neighbors (or $k - 1$ if k is odd). Then, with probability p, each edge (u, v) within

each of the '$n-$rings with k nearest neighbors' is replaced with a new edge (u, w) or (v, w), where w is a node of G chosen uniformly at random Newman [2010].

3.1.8 Sticky Graphs

A Sticky graph $G(n, m)$ has a degree function $d(i)$ and stickiness index function $s(i)$. The degree function $d(i)$ is the theoretical degree of node i. Note that this is not the same as the degree of node i in G. We define the stickiness index of node i as $\frac{d(i)}{\sum_i^n d(i)}$. An edge exists between two nodes u and v in G with probability $s(u)s(v)$. Sticky graphs were introduced by Pržulj and Higham [2006] to model biological networks and in particular, PPI networks. Sticky graphs attempt to capture the following two ideas: (i) proteins with high degree have binding domains which would be commonly involved in interactions, (ii) a pair of proteins is more likely to interact if they both have high stickiness indices. As we shall see, the STICKY model does a good job at describing existing PPI networks but cannot help us understand their evolution since it is built to duplicate the topology of an existing network.

3.2 Generating Synthetics Based on Theoretical Models to Match a Real-World Network

In general, given a real-world network $G_{real}(n, m)$, to generate a synthetic graph $G_{synth}(n_0, m_0)$, based on a graph model M (such as ER, GEO, SF, etc), we set the parameters required to generate a graph of model M so that G_{synth} matches G_{real} as closely as possible in size and density. To generate ER, SF, SW and GEO networks to match a specific real-world network, we set the relevant parameters for each model (such as number of nodes n, edges m and density p for the ER model, number of nodes n, edges m and radius r for the GEO model, etc) so that the resulting graphs match the size and density of the real-world network.

To create a GEOGD synthetic based on $G_{real}(n, m)$, the number of nodes required is n and number of edges is m. We can pick an arbitrary probability p and radius r and generate the GEOGD graph G_{synth} on n nodes as described earlier. The seed network used in the initialization should be a dense subgraph of the real-world network [Hormozdiari et al.]. However, the number of edges in G_{synth} may be greater or less than the required number m. To achieve exactly m edges, we adjust the value of r to r_0 as follows: compute and sort all the pairwise distances between the n nodes of G_{synth} and set r_0 to the r_0'th pair-wise distance Pržulj et al. [2010a].

To create an SFGD synthetic based on $G_{real}(n, m)$, the number of nodes required is n and the number of edges is m. We can pick arbitrary attachment and detachment probabilities p and q and generate the SFGD graph G_{synth} on n nodes as described earlier. However, the number of edges of G_{synth} may be different from the desired number m. In this case, we repeat the SFGD procedure, changing the value of q via a binary search on each repetition, until we have within 1% of edges (m) as in Pržulj et al. [2010a].

To create an ERDD synthetic based on $G_{real}(n, m)$, we use the stubs method to create G_{synth} as described in Milenković et al. [2008]. We assign each node of G_{synth} a stub number equal to $deg(n_i) \forall n_i \in V(G_{real})$. We then randomly select any two nodes of G_{synth} and if their stub numbers are greater than 0, we place an edge between those nodes and decrease the stub count of both nodes by 1. We repeat this procedure until each nodes of G_{synth} has a stub count of 0.

To create a Sticky synthetic based on $G_{real}(n, m)$, we create G_{synth} with n nodes and assign each node of G_{synth} a stickiness index according to the degree of the corresponding node in G_{real}, as described earlier. We place an edge between two nodes u, v in G_{synth} with probability equal to the product of their stickiness indices.

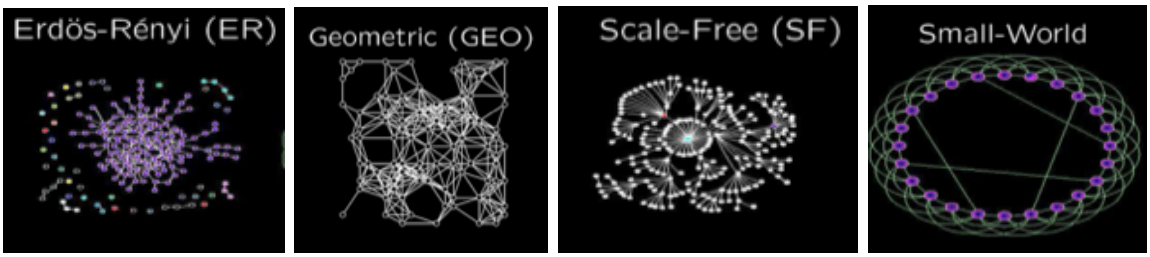


Figure 3.1: These images provide intuition on the structures of these four synthetic networks [Pržulj, 2005]. The edges of the ER graphs appear randomly placed. The regions in the GEO graph where nodes are physically close to each other are the densest and busiest regions in the graph. There are hubs in the SF graph in which there are a few nodes of high degree connected to many nodes of lower degree. Nodes in the SW graph are well-connected locally and require a few hops to reach a node outside of its locality.

3.2.1 Model-Driven Synthetics

Synthetic network models which describe the topology and inter-connectivity of the network are described to as *model-driven* synthetic networks. In particular, they tend to have only a few descriptive parameters such as the number of nodes and number of edges. ER, GEO, GEOGD, SF, SFGD, SW graphs fall into this category. For example, saying a graph is Geometric has implications on where the nodes might be positioned in space relative to one another. Figure 3.1 shows the distinct structures of ER, GEO, SF and SW graphs.

3.2.2 Data-Driven Synthetics

Some synthetic models require more input parameters to build a graph. Other than number of nodes and number of edges, these may include degree distribution or clustering coefficients. However, they may not provide as clear insight into the topology of the graph. For example, two graphs may have very similar degree distributions but their exact connections may cause them to look entirely different. Sticky and ERDD graphs can be described as data-driven synthetic models. A data-driven model may even be viewed as a hypothesis specifically designed to model a particular and very specific network structure. We may also describe

these models as *network-specific*. As described above, both Sticky and ERDD networks are created based on a *specific* input network whereas model-driven synthetics have more degrees of freedom because their input parameters are less restrictive.

3.3 Previous Models for PPI Networks

Networks have been used for decades to model biological processes and interactions such as proteome-scale interactions in human cells [Luck et al., 2017], gene regulatory networks [Karlebach and Shamir, 2008] and brain connectomes [Milano et al., 2017]. A protein-protein interaction (PPI) network is a graph whose nodes are proteins and edges represent observed physical interactions between the proteins (nodes) they connect. Many models have been suggested to better comprehend and describe the connections and patterns within PPI networks. Early work suggested that the degree distribution followed a scale-free law [Barabási et al., 2003, Salathé et al., 2005]. While evolution certainly must play a role in structuring biological networks [Wang and Zhang, 2007, Bianconi et al., 2009], different modules may show significantly different structures [Pinkert et al., 2010, Luo et al., 2006]. We readdress this question primarily using *graphlet* measurements. Although graphlets have been used previously to model PPIs [Pržulj and Higham, 2006, Hayes et al., 2013, Janjić et al., 2014], the amount of data available has increased dramatically in recent years, and so we revisit modeling the structure of these networks.

One of the earliest studies showed that the SFGD network model compared favorably with yeast by examining the degree and K-hop distributions, but did not test suitability with any other PPI network nor compared with other network models [Vázquez et al., 2003]. Using the RGFD network comparison measure, PPI networks of yeast and fly were shown to match GEO network models more than SF [Pržulj et al., 2004], and the STICKY model was shown to better replicate yeast, fly, worm and, human PPI networks compared to GEO, SF and

ER [Pržulj and Higham, 2006].

Several studies following seeked to show that PPI networks of various species (yeast, worm, fly and, human, amongst others) were (i) more similar to the structure of GEO networks [Pržulj, 2007, Higham et al., 2008, Pržulj et al., 2010b], or (ii) modeled well by STICKY graphs, often employing the GDDA measure to determine the quality of fit [Pržulj, 2007, Janjić and Pržulj, 2014, Janjić et al., 2014, Hayes et al., 2013]. Using GDDA, GEOGD and SFGD models were shown to outperform the fits of SF, ER, GEO and, ERDD models on PPI networks of yeast, fly, human and worm [Pržulj et al., 2010a]. This suggested that gene duplication was responsible for some of the underlying structure of PPI networks, not only the power-law of the SF model [Pržulj et al., 2010a].

The majority of studies on modeling PPI networks indicate that STICKY and GEO models are the best fits but only a few have compared against fits of GEOGD and SFGD models [Pržulj et al., 2010a, Hayes et al., 2013]. In addition, they have been tested primarily using GDDA and sometimes RGFD but the behaviours of these and other measures may be different from each other and have never been explored. Though over the past decade, PPI networks have been continuously updated [Stark et al., 2006, Breitkreutz et al., 2008, Chatr-Aryamontri et al., 2012, 2017], many are still incomplete, sparse, and have false positives and negatives [Vidal, 2016]. Furthermore, the studies performed thus far to model PPI networks as a whole, used older PPI network data.

3.4 Our Contribution

We re-evaluated these models on the newest data available from BioGRID[Chatr-Aryamontri et al., 2017] (downloaded September 2018) for 9 species. Table 3.1 contains the regular names of these 9 networks as well as their sizes. To the best of our knowledge this has not

yet been performed, as the data are relatively new. We examine the graphlet distributions of several models to distributions of the updated networks, and find that overall, other than the STICKY model, the SFGD and SF models outperform other traditional models (including GEO and GEOGD) in matching the structure of *all* these 9 BioGRID PPI networks. We analyze the fit of the network models using several measures that have been put forward in the past decade as being suitable to measure network distances (or similarities) - RGFD, GDDA, GK, SGK and, GCD. We also briefly discuss the variation in behavior of the network measures and their utility in assessing network similarity. While most measures agree that STICKY is usually the best-fitting model, the p-values for the STICKY fits are still many orders of magnitude from optimal.

Species (*Latin name*)	Code	Common Name	NumNodes n	NumEdges m	Density	Mean Degree
Saccharomyces cerevisiae	SC	Brewer's Yeast	6879	104719	0.00443	30.44
Schizosaccharomyces pombe	SP	Fission Yeast	2951	8754	0.00201	6.09
Drosophila melanogaster	DM	Fruitfly	8836	46288	0.00119	10.48
Homo sapiens	HS	Human	22376	277940	0.00111	24.85
Caenorhabditis elegans	CE	Roundworm	3276	5638	0.00105	3.46
Arabidopsis thaliana	AT	Thale Cress	9571	35253	0.00077	7.37
Rattus norvegicus	RN	House Rat	3569	4952	0.00046	2.80
Mus musculus	MM	House Mouse	12817	37915	0.00046	5.91
Escherichia coli	EC	Bacteria	2044	12800	0.000025	12.52

Table 3.1: BioGRID PPI networks [Chatr-Aryamontri et al., 2017] used in this study, ordered by edge density, downloaded in Sept. 2018.

3.5 Method

We created 500 synthetic versions of each of the 9 BioGRID PPI networks using each of the following models: ER, ERDD, SF, SFGD, GEO, GEOGD and, STICKY. To generate ER, SF and GEO networks, we set the relevant parameters (such as number of nodes, density, radius, attachment index, etc) so that the resulting graphs matched the size and density of each of the PPI networks. We generated 9 sets of SFGD networks for different values of p ranging from 0.1 to 0.9 in increments of 0.1. For each of these sets, we exactly matched the

number of nodes of the original PPI network, and did a binary search on the corresponding q (cf. §3.2) value until the synthetic graph contained within 1% of the number of edges in the real network, as in Pržulj et al. [2010a]. We generated GEOGD synthetic networks using both the expansion and probability cutoff methods described in Pržulj et al. [2010a], incrementing the probability by 0.1 from 0.1 to 0.9. Therefore, we created 500 networks from each of ER, ERDD, SF, GEO and, STICKY models, 4500 SFGD networks, and 5000 GEO-GD networks, for a total of $(2500 + 4500 + 5000) * 9 = 108,000$ synthetic networks.

3.5.1 Log Ratio Distribution and AUC

Suppose G is a network with graphlet distribution given by $f_G : \mathbb{N} \mapsto [0, 1]$, where $\sum_g f_G(g) = 1$. Then $f_G(g)$ represents the proportion of graphlets of type g present in G. Define $r_{(G,H)}(g) = log(\frac{f_G(g)}{f_H(g)})$ as the log ratio of the proportion of graphlet g in two graphs G and H. If $r_{(G,H)}(g)$ is positive (negative), then the proportion of graphlet g is higher (lower) in G than in H. Define Area Under Curve (AUC) $= \sum_g |r_{(G,H)}(g)|$. We call this quantity AUC because it quantifies the amount by which the log ratio deviates from a horizontal axis (0). For two very similar networks, r_g should be close to 0 for all graphlets, and hence, AUC should be close to 0.

We ran ORCA [Hočevar and Demšar, 2014] on all the above networks to count all of the graphlets of size $k = 2, 3, 4$, and 5. We examined and compared the log ratio graphlet distributions, AUCs and, computed RGFD, GCD, GDDA, and GK measures between the synthetic model networks and the original BioGRID networks, both to observe which models fit the best, and to compare the measures themselves to each other.

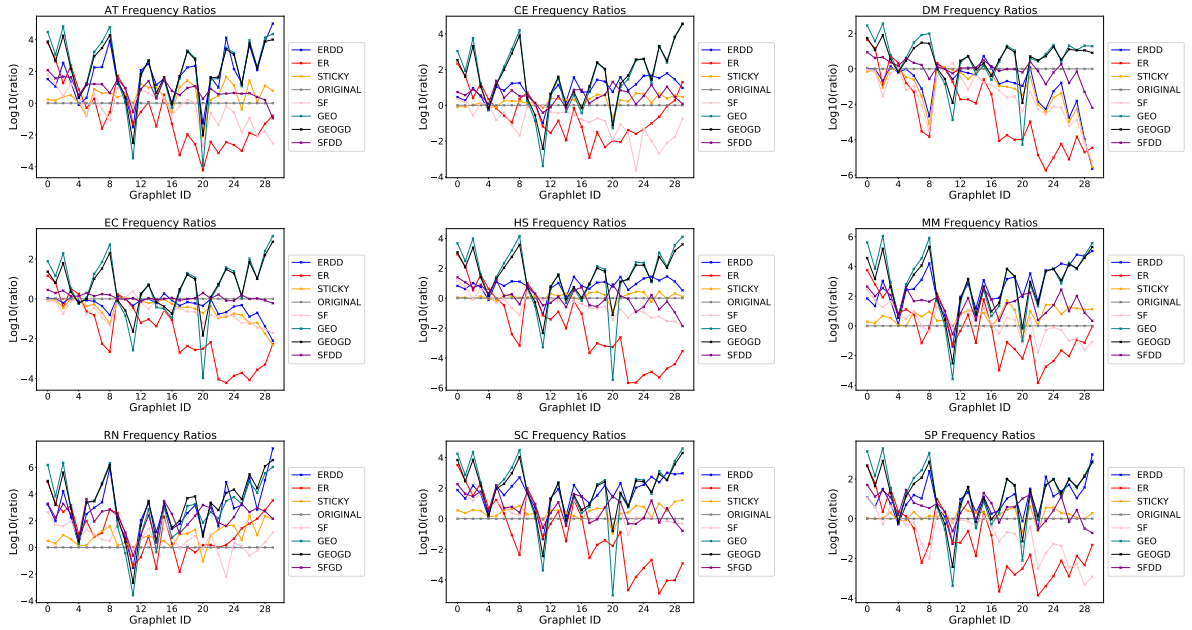


Figure 3.2: Each panel depicts the log ratio of graphlet counts between model and species, for each model (cf. §3.2), across all graphlets $g \in \{0..29\}$ from Figure 1.1. Each panel depicts all models across all graphlets for one species (cf. Table 3.1). The graphlet count for the model is taken from the mean across all 500 synthetic networks that were created for that species under that model. A ratio of 1 (and thus log of zero) depicts exact agreement in graphlet count between model and species for that graphlet. Thus, a good model will have a curve that remains close the x-axis across all graphlets. Visual inspection suggests that, for most species, the STICKY model remains closest to the x-axis, with SF and SFGD following close behind; this observation is quantified in Table 3.2.

3.6 Results

3.6.1 Assessment of Fits by Graphlet-Based Measures

Figure 3.2 shows the average log ratio $r_{(G,BioGRID)}(g)$ over 500 synthetic networks, for the 30 graphets, of sizes 2, 3, 4 and, 5, for each model. We observe that STICKY's log ratio graphlet distribution most closely matches the graphlet log-distribution of the BioGRID network, since its curve most closely 'hugs' the x-axis. After STICKY, on average SF and SFGD do reasonably well, whereas ER, ERDD, GEOGD and GEO are generally far from good fits (Figure 3.2). For each species, the differences in log ratios were very small (on order of magnitude 10^{-1}) within the 10 GEOGD and 9 SFGD types. Hence, we have plotted only the best performing GEOGD and SFGD models from among the GEOGD expansion and GEOGD probabilistic (p ranging from 0.1 to 0.9) networks and the SFGD models (p ranging from 0.1 to 0.9), respectively. Table 3.2 quantifies these observations with AUC values, corroborating the visual observation that STICKY is the best fit, followed by SFGD on most networks and SF on others.

Model \ Species	AT	CE	DM	EC	HS	MM	RN	SC	SP	Average
STICKY	**22.4**	**9.3**	36.2	16.4	**6.2**	**22.3**	**27.7**	**15.9**	**8.5**	**16.5**
SFGD	*27.9*	*17.4*	**12.3**	**4.00**	*18.0*	45.7	67.7	27.4	*21.7*	*24.2*
SF	*28.0*	29.4	46.9	20.7	21.7	**25.5**	**28.2**	*20.7*	30.7	*25.2*
ERDD	58.1	29.7	31.4	*11.2*	27.1	80.6	86.9	53.8	32.5	41.1
ER	53.4	34.2	79.5	56.8	77.0	47.0	47.7	62.0	52.1	51.0
GEOGD	70.7	55.4	*26.8*	30.1	51.6	87.5	101.8	62.9	43.7	53.0
GEO	76.8	58.2	35.1	37.27	62.3	92.0	96.4	71.1	49.0	57.8

Table 3.2: Quantifing the observations from Figure 3.2: Values represent the area between the curve and the x-axis, i.e. $mean(AUC(G, BioGRID))$. The models with smallest AUCs are those which most closely "hug" the x-axis in Figure 3.2. The rows are ordered best-to-worst according to the "Average" column. Within each column, the best value is boldfaced and the second-best value is italicized, with up to two if there are close ties. Note that since log is monotonic, the ordering would be the same even without the logarithm.

Figure 3.3 depicts the RGFD, GCD, GDDA, and GK scores between the model networks

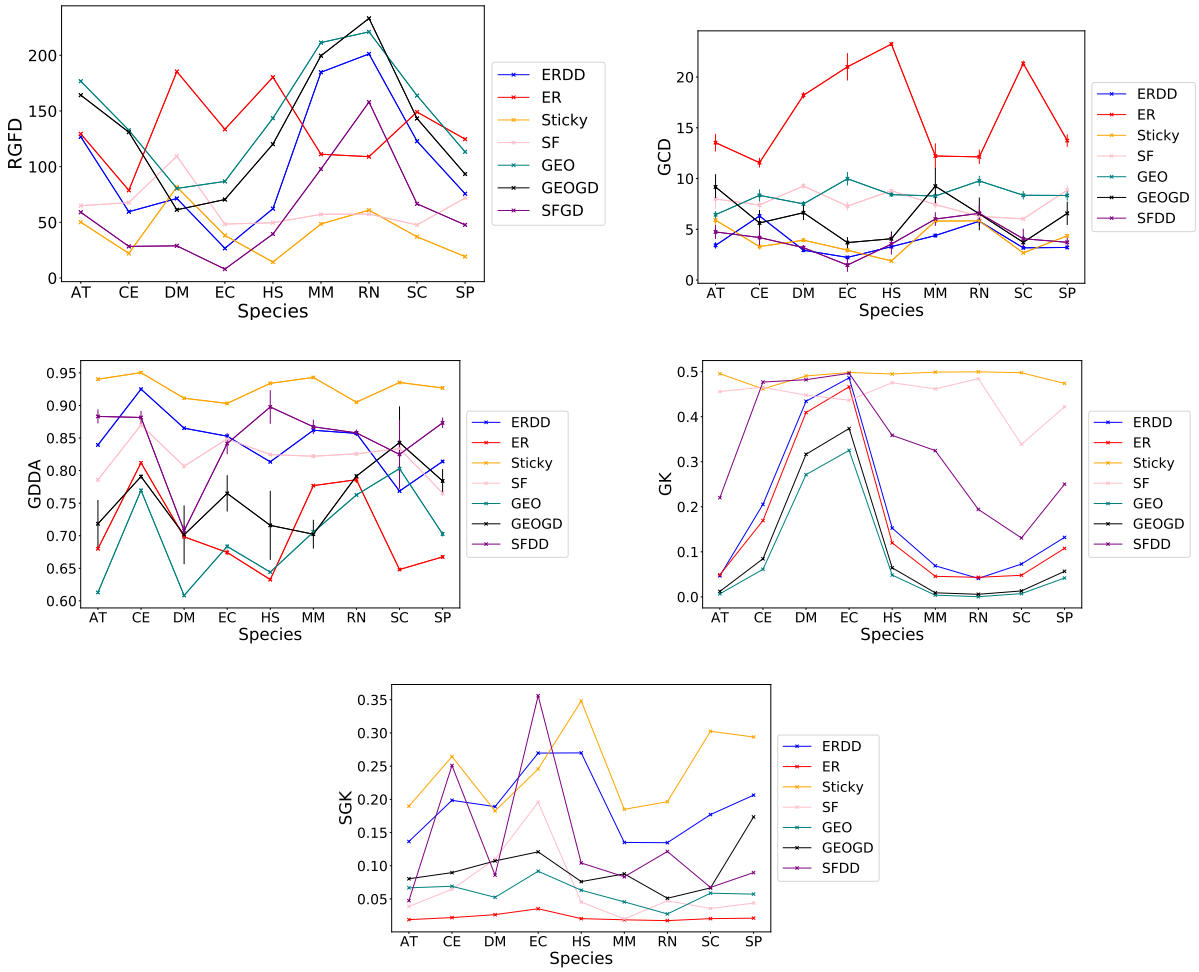


Figure 3.3: The average computed score comparing species-vs-model networks for the distance measures RGFD and GCD, and the three similarity measures GDDA, GK and, SGK. Error bars along each curve depict 1σ standard deviations of the species-vs-model scores across the 500 synthetic networks for that (species, model) pair. Note that all measures appear to agree that STICKY is the best model, being the lowest curve in the difference measures, and the highest curve in the similarity measures. Interestingly, the curves in the GK plot cross each other far less frequently than with the other measures, suggesting that GK maintains monotonicity in model quality, across all species, much better than the other measures; the significance of this is unclear.

and real network of each species. According to the RGFD and GK measures, other than STICKY, the graphlet topology of the 9 networks is fit best by SFGD and SF models. Using the GDDA and GCD measures, the best overall fits after STICKY are ERDD, SF and SFGD. All measures unanimously assess GEO, GEOGD and ER to be the worst fits. Using the four network comparison measures, CE, DM and EC are best matched (sometimes after STICKY) by the structure of SFGD networks. AT and MM appear to have some structural similarity to ERDD networks according to the GDDA and GCD measures. Older versions of all of these networks were already modeled well by STICKY graphs and we make the same observation here. This demonstrates further that the STICKY model is a plausible model for PPI networks even as the data evolve. We find that under GDDA, the order of suitability in modeling the structure of the species CE, DM, HS and SC networks has changed since the study of Hayes et al. [2013]. The four species were modeled best by STICKY, SFGD and GEOGD (in that order) but using the current data, in order of best fit, CE is best modeled by STICKY, ERDD and then SFGD; DM is best modeled by STICKY, ERDD and then SF; HS is best modeled by STICKY, SFGD and then SF; and SC by STICKY, SF and then SFGD. Both GDDA and GCD use graphlet orbital information, and they generally agree on the top three best-suited models but they often disagree on the ordering (agreeing in order only on HS). GK places STICKY followed by the SF model overall as most suitable to almost all the biological networks. We also find that the models most structurally similar to SC has changed from the study of Janjić et al. [2014]. Using the GDDA measure, the BioGRID SC in Janjić et al. [2014] was best modeled by STICKY, ERDD, then GEO, but we find that for SC, GDDA assesses SF and SFGD, as the best models after STICKY, and GEO is one of the worst fits. The high scores on most networks together with the overall best-fit results (STICKY, SFGD, SF, ERDD) may be suggesting that the PPI network structures are exhibiting patterns of more than one network model.

Fits by Senatorial Graphlet Kernel

Since Senatorial Graphlet Kernel is a much more stringent similarity measure, we observe that the best fits for the BioGRID networks are starkly different under this measure (see Figure 3.3. Firstly, no model scores particularly well for any species. STICKY receives moderate scores on HS, SC and SP, and SFGD receives a moderate score on EC. After STICKY, ERDD is the next best fit, but scores too low to be described as a good match to the networks. This trend is similar to that seen under GDDA, but not under the other measures. This coupled with the large variance seen amongst low scores may be due to the more stringent behaviour of SGK. SGK scores are higher if the graphlet counts in each network match for each graphlet type, and scores more strictly otherwise. The SGK value is like an average ratio of similarity across all graphlets. As with other measures, we observe that under SGK, the GEOGD, GEO and ER models perform worst in modeling the PPIs. The overall low scores under SGK may be suggesting something completely different from other measures - that none of these typical models are suitable for modeling current PPI networks.

3.6.2 Mixed Agreement Across Measures

One striking feature of all these network measures is that their behavioural trends do not match each other, even though they have similar input values and were all designed to quantify the similarity (or difference) in networks. Moreover, the best fits under GDDA and GCD do not agree with the models which had the smallest AUCs from Table 3.2, whereas GK and RGFD both agree with the AUCs. This is not so surprising because the AUC, RGFD and GK formulae are all similar to each other. Table 3.3 attempts to quantify the relationships between these network comparison measures with the Pearson correlation coefficients over all models and all species for each pair of measures . The p-values indicate

Measures	ρ	p**-value**
RGFD and AUC	0.998	3.2×10^{-78}
SGK and GDDA	0.621	5.5×10^{-8}
GK and GDDA	0.492	4.3×10^{-5}
GCD and AUC	0.459	1.6×10^{-4}
RGFD and GCD	0.477	7.7×10^{-5}
GK and SGK	0.45	1.8×10^{-4}
GK and GCD	-0.240	5.8×10^{-2}
GDDA and AUC	-0.522	1.1×10^{-5}
SGK and AUC	-0.536	5.8×10^{-6}
RGFD and GDDA	-0.541	4.8×10^{-6}
RGFD and SGK	-0.545	3.9×10^{-6}
GCD and GDDA	-0.614	8.6×10^{-8}
SGK and GCD	-0.620	6.0×10^{-8}
GK and AUC	-0.757	7.0×10^{-13}
RGFD and GK	-0.757	7.1×10^{-13}

Table 3.3: How the various graphlet measures agree with each other, measured by the Pearson correlation ρ of their model-vs-data similarity (or difference) scores. The magnitude of the correlation depicts "amount" of agreement; the sign indicates whether the measures agree in direction (i.e., both similarities or both differences) or are opposite type (difference vs. similarity). The p-value is the probability that the observed correlation is due to chance. Thus, RGFD and AUC are virtually identical, while GK and GCD have such a low correlation (0.240) that the p-value—just 5.8%—means their agreement is barely distinguishable from random.

that the coefficients are *highly* statistically significant. RGFD, GK and AUC have high pairwise correlations. GCD and SGK are very weakly correlated with both GK and AUC. The values of AUC, RGFD and GK may be influenced most by the graphlets with the highest counts. For example, two networks with counts of 1 and 2 of a specific graphlet, can be considered to be 50% different due to this graphlet, or can be considered to have a graphlet count difference of only 1 added to the rest of the differences. In the former case, the difference of one graphlet carries a bigger weight than the latter case, which weighs the 1 graphlet relative to the remaining graphlet count differences. The latter is the behaviour of AUC, RGFD and GK. GDDA has been previously argued to be a sensitive measure [Hayes et al., 2013]. Since GDDA, GCD and SGK all have fair correlation coefficient with each other (Table 3.3), it is possible that GCD and SGK are also sensitive to small differences in graphlet counts.

3.6.3 Precision/Recall of Measures

To evaluate the performance of the measures, we used a Precision-Recall (PR) curve. Network pairs that were generated from the same model are defined as True and networks generated from different models are defined as False. For a given threshold ϵ, we computed the number of:

1. True Positives (TP), i.e. the number of True pairs having pairwise similarity greater than ϵ.

2. True Negatives (TN), i.e. the number of False pairs having pairwise similarity less than ϵ.

3. False Negatives (FN), i.e. the number of True pairs having pairwise similarity less than of equal to ϵ.

4. False Positives (FP), i.e. the number of False pairs having pairwise similarity smaller than ϵ.

Precision is defined as Precision $= \frac{TP}{FP+TP}$ and recall is defined as Recall $= \frac{TP}{TP+FN}$. The PR curve is shown in Figure 3.4 for SGK, RGFD, GCD and GK. GK has by far the highest AUPR and SGK has the lowest.

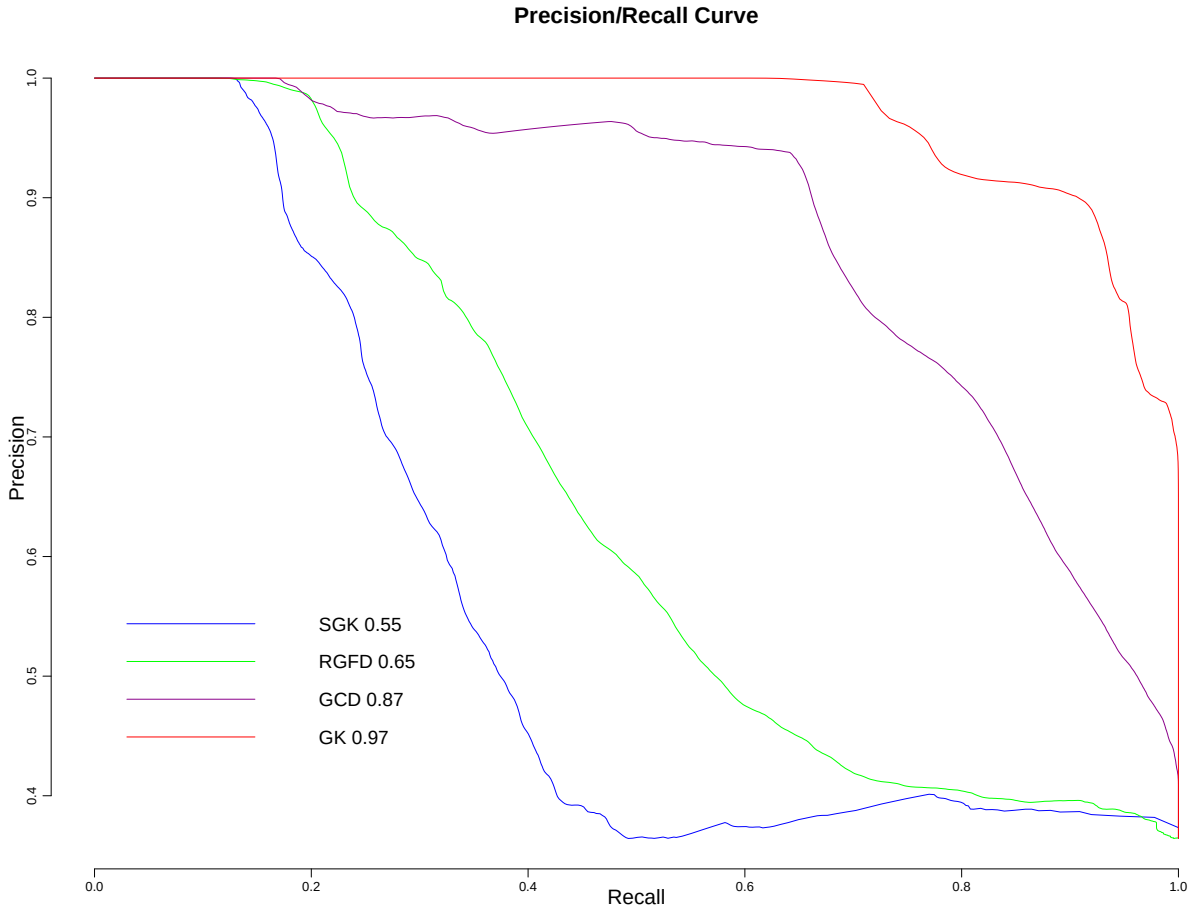


Figure 3.4: The Precision Recall curve for four measures are shown here. GK has the highest AUPR and SGK has the lowest.

Species	Graphlets (best,worst)	Mean STICKY Count	σ of STICKY Count	BioGRID Count	BioGRID Distance to STICKY Mean
CE	0	5619	74	5638	0.26σ
	20	5572	886	21554	18σ
SP	27	33810	3057	32556	0.41σ
	20	62298	5221	330540	51.4σ
SC	6	6.18×10^8	6.09×10^6	618385434	0.06σ
	20	5.20×10^7	940839	1833805726	$\mathbf{1900}\sigma$
EC	0	12784	110	12800	0.15σ
	29	14.6	8.85	2727	306σ
RN	25	3188	334	2945	0.73σ
	20	1405.2	218.206	53753	240σ
AT	21	2.87×10^7	912783	29801214	1.15σ
	20	1.18×10^6	54516	256721361	$\mathbf{4700}\sigma$
DM	0	46268	206	46288	0.09σ
	29	4.29	3.79	1278120	$\mathbf{337{,}000}\sigma$
HS	0	277199	514	277940	1.44σ
	20	1.86×10^8	2.75×10^6	1369437894	430σ
MM	0	37650	188	38075	2.26σ
	20	730868	34637	9773813	261σ

Table 3.4: The mean and standard deviation of each graphlet's count in the STICKY synthetic networks, together with the deviation of the true count from the synthetic mean. For each species, the best-fitting and worst-fitting graphlets are shown; the absolute worst offenders (thousands of standard deviations away) are highlighted in bold. Though STICKY is generally the best structural match among all 7 models explored, its fit often deviates by several orders from the current BioGRID network. The BioGRID network has overall the highest standard deviation on graphlet G_{20} from the STICKY synthetics across all species.

3.7 Evaluation of STICKY as Best Fit

As STICKY is the overall best fit of the PPI networks, we further examine its fit to the BioGRID networks. For each species, and for each graphlet type, we computed the mean and standard deviation of the graphlet count across all 500 synthetic STICKY networks. BioGRID's graphlet count is shown in Table 3.4 as a multiple of the standard deviation away from the mean count in the synthetics. This factor is small for graphlet 0 (the number of edges in the graph). However, the factors grow very large on some graphlets in each species (more than $10^5\sigma$ as is the case for DM), as shown in Table 3.4. This means that though STICKY is the best amongst the models examined in this study, it is still a distant match from the current versions of PPI networks.

3.8 Discussion and Conclusion

We have performed a comprehensive evaluation of the fit of the 9 major BioGRID networks to several network models across a wide range of graphlet measures. While there is significant disagreement between measures as to rank the models (as explicitly shown by the range of Pearson correlations between measures tabulated in Table 3.3), the gross conclusion is that the STICKY model is the best-fitting model to BioGRID PPI networks. This is not surprising since the STICKY model was specifically created to model the affinity of protein pairs to interact. The stickiness index is closely related not only to the degree-distribution, but to the *inter-node* topology of the network. It preserves graphlet structure better than most generic models. STICKY has remained the best-fitting model throughout 10 years of increasingly voluminous PPI network data. However, we turn to other models to gain additional insight into the inter-connectivity and structure. Previous studies [Pržulj et al., 2004] suggested that since yeast was geometric in structure (at the time the study was

conducted), the resulting network's degree distribution would be Poisson and hence, would have a peak at the mean degree. However, the current yeast network's degree distribution is far from Poisson, as the peak is at 1 (757 nodes with degree 1) but only 85 nodes have the mean degree of 25. We find that it is not GEO, but SFGD and SF models that score well according to a few network measures, suggesting that current PPI networks are partially scale-free in nature and that gene-duplication also has a role to play in their structure. As seen in Figure 3.3 and Table 3.3 the measures have significant disagreement on the ordering of models from best to worst. The models stipulated as the best-fits for PPI networks have changed as PPI data have been updated. We have also observed that the best-fitting model may depend on the comparison measure used for the assessment. Further, the 'best' fit may not be the *right* fit as we have seen with the graphlet count deviations in Table 3.4. It is plausible that STICKY *is* right model for PPIs and the low p-values of Table 3.4 can be blamed not on the model, but on the noise and incompleteness of the data; this hypothesis will need to wait several more years to be more thoroughly tested.

Finally, the distinct disagreements between different graphlet measures depicted in Figures 3.2–3.3 and Tables 3.2–3.4 suggest that we are a long way from knowing the best way to use graphlets to measure topological similarities and differences. We are led to ask: what makes a measure suitable for comparing (biological) networks? What does it mean for two networks to be similar or different? We leave exploration of these questions as future work.

Chapter 4

Counting Graphlets and BLANT

Results in the previous chapter used ORCA, which exhaustively enumerates all graphlets, which is already infeasible on some networks and will become increasingly burdensome as biological networks continue to grow. Many existing methods that use graphlets for *any* purpose, first require an exhaustive enumeration of all graphlets either in the complete network being analyzed [Pržulj et al., 2004, Pržulj, 2007, Yaveroğlu et al., 2014, Ali et al., 2014], or all graphlets that touch a particular node [Rahman et al., 2014]. In either case, exhaustive enumeration of graphlets takes time that is exponential in the size of the network, the size of the graphlets enumerated and the largest degree of the network (see Equation 1.1). Such exhaustive enumeration is infeasible even on existing large networks, and the problem will only get worse as networks grow in size. While there have been significant improvements in the efficiency of exhaustive graphlet enumeration over the years [Pržulj et al., 2006, Marcus and Shavitt, 2012, Hočevar and Demšar, 2014, Melckenbeeck et al., 2016, 2017a, Rahman et al., 2014, Hočevar and Demšar, 2017], the sheer number of graphlets to enumerate means that no algorithm will ever be able to exhaustively enumerate all graphlets in a large network in a reasonable amount of time. It can take many hours to count all the graphlet-orbits of even a moderately-sized network such as HSapiens of BioGRID Chatr-Aryamontri et al.

[2012] and days for slightly larger networks such as Autonomous Systems in the SNAP database Leskovec and Sosič [2016].

A feasible alternative to exhaustive enumeration of all graphlets is to statistically sample them. This has been explored in Pržulj et al. [2006] and GRAFT Rahman et al. [2014] for up to 5-node graphlets. The former limits the time spent counting graphlets at each node in the network but made no effort to account for any bias introduced by limiting the graphlet-search time, and also analyzed the performance only on synthetic networks based on different models as well as on protein-protein interaction networks. GRAFT samples by selecting an initial set of edges and matches each sampled edge with each edge in each graphlet, enumerating all embeddings where such a match exists. This technique still has an exponential run time, especially for any edges sampled from dense regions of the graph.

Several other methods have been developed to estimate graphlet counts or graphlet concentrations. Seshadhri et al. [2013], Ahmed et al. [2014] estimated the number of triangles (Graphlet 2) in a network and Jha et al. [2015] estimated the number of 4-node graphlets. Using a set of uniformly sampled edges from the graph stream, Wang et al. [2016] estimated the number of any k-graphlets. There has also been a body of work utilizing random walks Bhuiyan et al. [2012], Wang et al. [2014], Ahmed et al. [2015], Chen et al. [2016] which have been quite successful in estimating graphlet counts and concentrations. Using the Metropolis-Hastings Algorithm as a basis, Bhuiyan et al. [2012] estimates $3, 4, 5$-node graphlet statistics altogether but lacks speed due to rejection of samples; Wang et al. [2014] shows that a pairwise subgraph random walk (PSRW) outperforms the simple random walk (SRW) in estimating graphlet statistics; Chen et al. [2016] generalizes the the idea of Wang et al. [2014], computes the graphlet-state network on the fly, removes the bias in the concentration of graphlets sampled and provides a bound on the number of samples required using the Chernoff-Hoeffding inequality. Using Graphlet Kernel, Chen et al. [2016] also shows that the graphlet concentrations can be used to assess the similarity between networks but used

only a handful of networks.

To date, all methods for graphlet enumeration and sampling have been restricted to graphlets of size 5 or less, with the sole exception being JESSE [Melckenbeeck et al., 2016]. However, JESSE is also an exhaustive enumeration tool. Thus, no fast, general tool exists for sampling graphlets of size greater than 5. As Figure 1.3 shows, the number of graphlets (and graphettes) increase exponentially with k. The sheer number of graphlets and orbits for $k > 5$ may have been an impediment in developing graphlet counters and samplers for larger graphlets. Though [Yaveroğlu et al., 2014, Kuchaiev et al., 2010] have shown that $2, 3, 4$ and, 5-graphlets can extract a lot of information from networks, we cannot really conclude that graphlets of up to 5 nodes are sufficient for graphlet analysis on networks, without actually investigating and observing what larger graphlets can tell us about the topology of networks. In addition, though many sampling techniques have low errors in estimating the graphlet concentrations of networks, it has not been explicitly shown that they can reproduce the network comparison results obtained from exhaustive enumeration such as the results of [Yaveroğlu et al., 2014].

4.1 Our Contribution

We demonstrate that statistical sampling of even a tiny sample of graphlets, taking mere seconds of CPU time, can effectively reproduce network comparison results that would require days, weeks, or months of CPU time using existing exhaustive enumeration methods. We leverage the algorithm of Hasan et al. [2017] together with the BLANT tool of Hayes and Maharaj to survey and implement sampling algorithms - Edge Based Expansion Sampling [Hayes (personal communication)], Node Based Exansion Sampling [Hayes and Maharaj], Neighbor Reservoir Sampling and the algorithm of Chen et al. [2016] which we will refer to as MCMC (Markov Chain Monte Carlo). We show that each technique is able to repli-

cate the graphlet distribution of networks with some success. We also convert the sampled graphlets into k-graphlet orbit degree matrices for $3 \leq k \leq 7$ and make network comparisons using GCD to show that even a set of sampled graphlets is enough to distinguish between networks of different types and structures.

4.2 Different Graphlet Sampling Algorithms

We describe Node Based Expansion (NBE), Edge Based Expansion (EBE) [Hayes (personal communication)] and Neighbor Reservoir Expansion Sampling (NRE) in the following sections.

Algorithm 1 Helper Functions for Sampling Methods

```
procedure RemoveFromCandidates(Candidates, Neighbors)
    for each neighbor neigh in Neighbors do
        Remove neigh from Candidates
    end for
end procedure
procedure AddToCandidates(Visited Candidates, Neighbors)
    for each neighbor neigh in Neighbors do
        if not Visited(neigh) then
            Add neigh to Candidates
            Visited ← neigh
        end if
    end for
end procedure
```

4.2.1 Node Based Expansion (NBE)

We go over the details of Node Based Expansion [Hayes and Maharaj]. When sampling a single graphlet, during the node selection process, we keep a set of the nodes visited, called $Visited$, a set of candidate nodes, $Candidates$ and a set of nodes already selected for the graphlet construction, S. First, we choose an edge $e = (v_1, v_2)$ from $G(V, E)$ at random.

The initial choice of an edge is to increase the chances of sampling from the denser regions of the network. The vertices $v1$ and $v2$ are put into S. We expand the nodes at $v1$ and $v2$ and add all of the previously unseen nodes that are discovered into $Candidates$ and $Visited$. We then randomly select a node from $Candidates$ to add to S. We repeat this procedure of expanding the selected node and adding its neighbors to $Candidates$ until either $|S| = k$, or until $Candidates$ is empty but S is not yet fully constructed. In the former case, we build g from S and return g but in the latter case, we select a previously unseen node v_{rand} from G, add v_{rand} to S and to $Visited$, add $N(v_{rand})$ to $Candidates$ and repeat the above procedure. Algorithm 2 contains details and pseudocode of the sampling method.

Algorithm 2 Node Based Expansion

procedure NODEBASEDEXPANSION(Graph $G(V, E)$, graphlet-size k)
  $S \leftarrow \emptyset$
  $Visited \leftarrow \emptyset$
  $Candidates \leftarrow \emptyset$
  Select an edge $e = (v_1, v_2)$ uniformly at random from E
  Add v_1 and v_2 to S
  $Visited(v_1) \leftarrow True$
  $Visited(v_2) \leftarrow True$
  AddToCandidates($Visited$, $Candidates$, $N(v_1) \cup N(v_2)$)
  while $|S| < k$ **do**
    if $|Candidates| > 0$ **then**
      Select a node v_{rand} uniformly at random from $Candidates$
      Remove v_{rand} from $Candidates$
    else
      Select a random node v_{rand} from $G - Visited$
      $Visited(v_{rand}) \leftarrow True$
    end if
    Add v_{rand} to S
    AddToCandidates($Visited$, $Candidates$, $N(v_{rand})$)
  end while
  Build g from S
  return g
end procedure

Complexity

The sampling method initially selects an edge and hence two nodes to add to the sampled graphlet. All the neighbors of these two initially selected nodes is added to the Candidate set. This has an average run time of $O(d)$, where d is the average degree in the network. For each of the remaining up to $k-2$ spots left to fill in the sampled graphlet, a node is selected at random either from the Candidate set (if it is not empty) or from the network and all its neighbors are added to the Candidate set. This has a worst case run time of $O((k-2)d)$. It takes $O(1)$ time to determine the canonized version of the sampled graphlet as well as its associated orbits, because BLANT uses a lookup table [Hasan et al., 2017]. Thus, if m samples are to be taken, the worst case runtime of this method is $O(md(k-1))$.

4.2.2 Edge Based Expansion (EBE)

In this method, we do not explicitly maintain *Candidates*, but instead keep only the *count* of *all* edges (including internal ones because there is no O(1) way to exclude them) that emanate from all nodes $v \in S$ as S is being built. Then we pick one such edge uniformly at random and discard it if it leads back into V. This method has the advantage that it is asymptotically faster than the Node Based Expansion method as the mean degree gets large, but has two disadvantages: first, it may be slow if the mean degree is small, because most randomly chosen edges will be discarded as internal; and more importantly, we cannot easily detect if the number of edges leaving S is zero. Although we do not show it in the pseudocode of Algorithm 3, we need to detect this and fail (or retry) if the number of trials in the outer **while** loop exceeds some maximum.

Algorithm 3 Edge Based Expansion

```
procedure EdgeBasedExpansion(Graph G(V, E), graphlet-size k)
    Select an edge e = (v_1, v_2) uniformly at random from G
    S = {v_0, v_1}
    // A is an ordered array of nodes in S in step i
    A[0] = v_0
    // C is an array of the cumulative node degree in A
    C[0] = Degree(v_0)
    A[1] = v_1
    C[1] = C[0] + Degree(v_1)
    D = C[1]
    while |S| < k do
        pick integer j uniformly at random ∈ [0, D)
        i = 0
        while C[i] ≤ j do
            i++
        end while
        h = j − (C[i] − Degree(A[i]))
        v =AdjList(A[i])[h]
        if v ∉ S then
            C[|S|] = C[|S| − 1] + Degree(v)
            A[|S|] = v
            S = S ∪ {v}
            D+ = Degree(v)
        end if
    end while
    Build graphlet g from S
    return g
end procedure
```

Complexity

For each node we select uniformly at random from the outset of S, we find the node via the degree cumulative array C in $O(d)$ time. Since we need to add k nodes into S, the run time is $O(kd)$ but since $k << d$ (i.e. the number of nodes in a graphlet is much less than the mean degree of nodes in the graph), the run time is $O(d)$.

4.2.3 Neighbour Reservoir Sampling (NRE)

This method is similar to Node Based Expansion (NBE) described above but attempts to reduce the bias of selecting denser graphlets and reduce the bias of selecting graphlets with a high degree node. First, we select g_0, an initial k-graphlet using NBE. During the next $r * k$ steps, we select a node v uniformly at random from $Candidates$. With probability $\frac{k}{r}$, we decide to insert v into g_{r-1} by deleting a node u from g_{r-1} selected uniformly at random. If the resulting graphlet g_r formed from $V(g_{r-1}) - \{u\} + \{v\}$ is connected, we keep it. Otherwise, $g_r = g_{r-1}$. v is added to $Visited$ and we update $Candidates$ to contain those nodes neighboring any node of g_k. The sampled graphlet is g_r. The number of steps r, after the initial graphlet is selected, is an important parameter in achieving a more accurate distribution of graphlets. If we set a finite value for r, then we stop swapping nodes into the graphlet after a exactly r steps. If we set r to infinity, then we stop swapping nodes into the graphlet only after the $Candidates$ set becomes empty. Algorithm 4 contains details and pseudocode of this sampling method. The idea of Neighbour Reservoir Sampling is inspired from the graph sampling Neighbour Reservoir of Lu and Bressan [2012]. However, while they determine the new node by uniformly at random selecting an edge connecting the nodes of the currently sampled graphlet g_{iter} to the rest of the unexplored network, we select the new node from our $Candidates$ set, which allows us to select the next node uniformly at random (i.e. nodes of higher degree are selected with the same probability as nodes of lower degree).

In addition, Lu and Bressan [2012] does subgraph sampling, i.e. they sample one subgraph to represent the entire network and hence, their r is always 'infinite'. We have modified this by experimenting with the number of steps to take before keeping a graphlet.

Algorithm 4 Neighbour Reservoir Expansion

procedure NEIGHBOURRESERVOIR(Graph $G(V, E)$, graphlet-size k, steps r)
  $S \leftarrow \emptyset$
  Sample k-graphlet g_0 from RVE
  $Visited$ initialized from RVE
  $Candidates$ initialized from RVE
  $iter \leftarrow 1$
  while $|Candidates| > 0$ and $iter \leq r * k$ **do**
    Remove v uniformly at random from $Candidates$
    $Visited(v) \leftarrow True$
    Generate p uniformly at random from $[0, 1)$
    if $p < \frac{k}{iteration+k}$ **then**
      Select a node u uniformly at random from g_{iter-1}
      Exchange u with v in g_{iter-1} to form g_{iter}
      if g_{iter} is connected **then**
        $outset \leftarrow N(u) - \cup_{i=0}^{k} N(u_i) \cap N(u), u_i \neq u, u_i \in g_{iter-1}$
        RemoveFromCandidates($Candidates$, outset)
        AddToCandidates($Visited$, $Candidates$, $N(v)$)
      else
        $g_{iter} \leftarrow g_{iter-1}$
      end if
    end if
  end while
  return g_r
end procedure

Complexity

The initial choice of a graphlet using NBE is $O(d(k+1))$. For a finite r, we take at most r steps to complete the graphlet formation. We expect to check for connectivity of g_{iter} and update the $Candidates$ set $\sum i = 1^{i=r} \frac{k}{(i+k)} < \frac{kr}{max(r,k)} = a$ times. Checking for connectivity takes $O(k)$ times for a k-graphlet and updating the candidate set takes on average $O(d)$ time. This takes $O(md(k+1) + mra(k+d))$ for m samples.

4.2.4 MCMC

We do not provide the explicit algorithm of Chen et al. [2016] here but refer the reader to their paper for a clear description of their method. In brief, a sliding window of l d-graphlets ($d < k$) is kept during the random walk on the network. Whenever the l d-graphlets contain k distinct nodes which form a connected graphlet, the graphlet concentration is updated appropriately using the stationary distribution for a random walk on the graph consisting of d-graphlets.

4.2.5 Properties of Node Based Expansion

Lemma 1. *Let $k > 1$ and $G(V, E)$ be a graph with at least k nodes and at least one connected k-graphlet. Each k-graphlet in G has a non-zero probability of selection.*

Proof. Consider any connected k-graphlet g consisting of nodes $\{v_1, v_2, ..., v_k\}$ and e edges in G. Since the sampling method randomly selects an edge in G to expand, Pr(intial edge selection is in g) $= \frac{e}{|E|} > 0$. If $k = 2$, then we have shown that each edge has a non-zero probability of selection. Suppose $k > 2$. Without loss of generality, suppose edge (v_1, v_2) in g is initially selected. Since g is connected, at least one of $\{v_2, ..., v_k\}$ is added to the Candidate set. Let S_i be the set of nodes which were already selected by the i^{th} step, $i \leq k$. S_i contains at least one node from g. The probability that the next node selected is from g is at least $T_i = \frac{1}{|\cup_{v \in S_i} N(v)|}$ with equality if there is exactly one vertex in S_i from g. Note that this probability is non-zero. Thus Pr(selecting g) $\geq \prod_{i=1}^{k-1} T_i$. Hence, each k-graphlet has a non-zero probability of selection in any sample. □

Corrollary 2. *In a connected graph $G(V, E)$, where $|V| \geq k$, if m samples are selected, then for each node v in G, $\lim_{m\to\infty} \Pr(v \text{ is selected}) = 1$.*

4.3 Features of BLANT

The *Basic Local Alignment for Networks Tool* (BLANT) was introduced in Hayes and Maharaj. BLANT does for networks what BLAST does for genomic sequences: efficiently create a graphlet database index that allows for fast search, comparison, and local alignment of graphlets from one or more larger networks.

4.3.1 Determining Graphlet and Orbit Type

Since we utilize the algorithm of Hasan et al. [2017], we use the binary representation of the sampled graphlet and determine the graphlet-type and respective orbits, in the look-up table, in constant time. This is one of the major contributing factors to sampling speed when compared with most other sampling algorithms. The user specifies the desired output format of the samples as our tool can produce the sampled graphlet count for each graphlet, sampled graphlet orbit signature matrix, and the list of all graphlets sampled with each graphlet listed in their canonical order of nodes. The above sampling methods can be modified in various ways to include samples of disconnected graphlets. This may be useful as Shervashidze et al. [2009] has already suggested that classification of graph types can be up to 10% more accurate when including disconnected graphlets. We leave further investigation in this direction as future work.

4.3.2 Parallelization

BLANT provides the option to do multi-threaded sampling, thereby allowing for a further speed-up in sampling graphlets.

4.3.3 Signature Matrices

Suppose there are $|V|$ nodes in graph $G(V, E)$, m samples of k-graphlets taken from G and T total orbits of k-graphlets. BLANT constructs the graphlet signature matrix $GSM(G)_{|V|\times T}$, where $GSM(G)_{[i,j]}$ is the number of sampled graphlets of type j that node i touches. Similarly, it can construct the graphlet orbit signature matrix $OSM(G)_{|V|\times T}$, where $S(G)_{[i,j]}$ is the sampled number of orbits of type j that node i touches in graph G.

4.4 Description of Data

To show that graphlet samples can capture sufficient information about the topology in different network models, we assembled a variety of network databases, including both synthetic networks of various network models as well as real world networks of different types.

4.4.1 Synthetic Networks

We generated both sparse and dense synthetic networks as described below.

Sparse Synthetic Networks

We generated synthetic networks on four different commonly used network models: ER, GEO, SW, and SF. For each of these models, we created 30 graphs each of 1000, 2000, 4000 and 6000 nodes and densities 0.01, 0.005 and 0.0075 as in Yaveroğlu et al. [2014] in order to replicate the properties of size and (sparse) density of real world networks. In addition, 100 Sticky graphs were also generated. Thus, we created $(4 \times 3 \times 4 \times 30) + 100 = 1540$ synthetic networks of various models, sizes and densities.

Dense Synthetic Networks

We also generated denser synthetic networks having a wide variety of parameter values. We created 400 synthetic networks - 100 from each of the same four models: ER, GEO, SW and SF. The number of nodes in each of these networks was selected uniformly at random between 500 and $10,000$ and the density was selected uniformly at random between 0 and 1.

4.4.2 Real World Networks

We also evaluated the performance of sampled GCD's on different types of real-world networks: 733 Autonomous Systems networks taken from the Stanford SNAP database Leskovec and Sosič [2016] representing routers of the Internet which communicate with each other based on the Border Gateway Protocal logs, 595 Chemoinformatic Enzyme networks representing atoms in molecules from Network Repository Rossi and Ahmed [2015b], 98 Facebook networks Rossi and Ahmed [2015b], 271 Combinatorial problem networks from The University of Florida Sparse Matrix Collection Davis and Hu [2011] and 535 Gene-μRNA networks Tokar et al. [2017] and 39 Brain networks from Nooner et al. [2012].

4.5 Experiment

We sampled m k-graphlets from each network in our databases of Sparse Synthetic networks, Dense Synthetic networks and Real World networks for $3 \leq k \leq 7$, and for $m \in \{10^6, 10^7\}$, using EBE, NBE, NRE and MCMC. We compared the distribution of graphlets accrued from each sampling method using the graphlet log-ratio plots described earlier in Section 3.5.1. We also obtained the graphlet orbit signature matrix option from BLANT and computed the Graphlet Correlation Distances between all pairs of networks in each database. Using these

pairwise distances, we employed Multi-Dimensional Scaling (MDS) to project each graph as a point into space, approximately preserving their relative distances from each other.

4.5.1 k-Graphlet Correlation Distance

Similar to the Graphlet Correlation Matrix of Yaveroğlu et al. [2014], we define k-Graphlet Correlation Matrix (GCM) $M_{T\times T}$ is constructed using correlations between graphlet counts (columns of $GSM(G)$) or graphlet orbit counts (columns of $OSM(G)$). The k-Graphlet Correlation Distance between two graphs G and H, $dist(G, H)$, is computed by taking the Euclidean distance of their respective correlation matrices. Note that this definition of GCD uses a specific value of k because our sampling is done for a specific value of k. Previous literature Yaveroğlu et al. [2014] uses GCD for all $k \in \{2, 3, 4 \text{ and, } 5\}$.

4.6 Results

4.6.1 Distribution, Log Ratio Distribution and AUCs

Figure 4.1 shows the average proportion of every $3, 4$, and, 5-graphlet in the sparse synthetic network database as counted by ORCA, as well as the average proportion of the same graphlets as sampled by EBE, NBE, Neighbor Reservoir Sampling with $r = 8$ and MCMC Sampling. The error bars indicate the standard deviation of each graphlet count within the original networks (shown on the ORCA lines) and within the sampled proportions. It is clear that NRE and MCMC samples capture the underlying distribution of graphlets in the networks much more correctly than EBE and NBE.

We observer clearer distinction in performance of the sampling algorithms by looking at their log ratio plots in Figure 4.2 and Figure 4.3. Figure 4.2 shows the average log ratio of graphlet

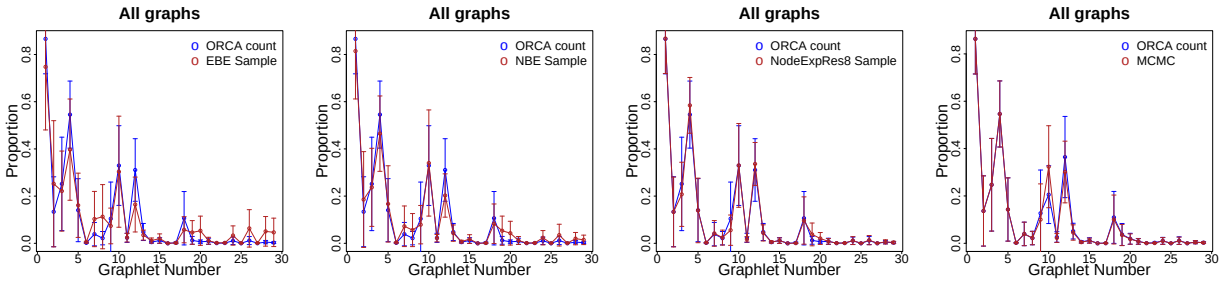


Figure 4.1: The figures above represent from left to right the proportions of sampled graphlet counts from EBE, NBE, NRE, and MCMC, as well as the exhaustive graphlet count proportion from ORCA, averaged over all synthetically generated ER, GEO, SF, SW and Sticky networks of various sparse densities. The error bars show the standard deviations of graphlet counts over all the networks. The error bars from all sampling methods roughly match the intrinsic variation of the synthetic networks themselves, as shown by the error bars from the ORCA counts. In addition, NRE and MCMC sampling appear to best match the original proportions of graphlets.

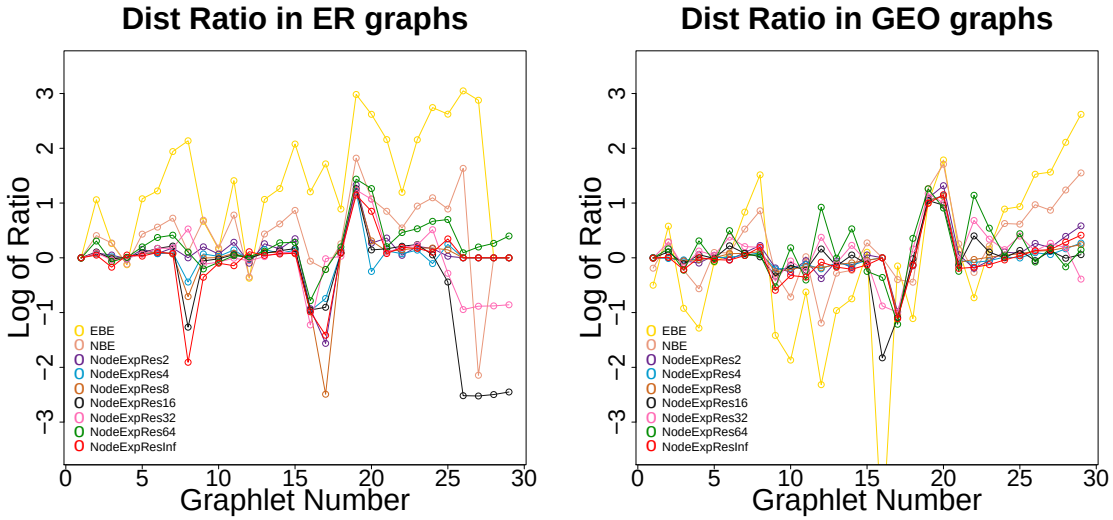


Figure 4.2: The figures above represents the mean log ratio of sampled graphlet counts to exhaustive graphlet counts over all synthetically generated ER and GEO graphs of various sparse densities. Neighbor Reservoir Sampling most closely 'hugs' the x-axis compared to EBE and NBE. Empirically, it is best for r values of 4 and 8.

Network Type \ Sampling Method	EBE	NBE	NRE-2	NRE-4	NRE-8	NRE-16	NRE-32	NRE-64	NRE-Inf	MCMC
GEO	33.7	16.3	7.2	**5.4**	**5.7**	7.7	9.7	11.2	7.3	**0.9**
ER	41.1	18.1	7.0	**5.4**	7.3	16.8	10.0	9.9	8.6	**4.8**
Sticky	39.8	19.8	19.3	18.6	**8.0)**				20.3	**1.7**
SW		24.1	13.2	9.3	**8.1**				10	1.5
SF	19.2	18.7	20	18.7	14.8	13.4	**10**		10.9	**3.4**

Table 4.1: The mean AUCs across all types of synthetic networks and all sampling methods are shown here. Neighbor Reservoir Sampling is the best from amongst the sampling methods which do not explicitly use any type of random walk across the network. In particular, an r value of 4 or 8 is emprically shown to be best for NRE. However, the MCMC is by far the best method for estimating the concentration of graphlets in a network. Its error is low on almost all networks. *Note that missing values will be filled in later.*

counts from EBE, NBE, and NRE sampling to the graphlet counts from ORCA for $k = 3, 4, 5$, on the synthetically generated ER and GEO networks. It is clear that EBE has the greatest sampling error, followed by NBE. NRE performs the best. In particular, we observe that the error in the log ratio plots is lowest when $r = 4$ or 8. We observe similar trends on other synthetic networks. It is clear that of these three methods, Neighbor Reservoir Sampling is the least biased. Table 4.1 quantifies the AUC of the plots in Figure 4.2.

Figure 4.3 shows the log-ratio of $3, 4, 5$-graphlets sampled by MCMC to ORCA on different types of synthetic graphs. We are unsure whether the consistent 'error peaks' observed at graphlets 10 and 23 are due to higher error for low-frequency graphlets or a bug in BLANT's implementation of MCMC and this will be investigated futher. However, it is clear that MCMC is by far superior to the previously mentioned sampling methods in determining the correct concentrations of each graphlet type present in the network, as it much more closely 'hugs' the x-axis. The AUCs for the MCMC Method are quantified in Table 4.1.

4.6.2 Graphlet Correlation Matrix Corrplots

The Graphlet Correlation Matrices have distinct correlation patterns for different kinds of networks. Figure 4.4 visualizes the GCM for an SF and GEO network, from both NBE and MCMC sampling of 5-graphlets, and from ORCA (i.e. using a full graphlet count). We see

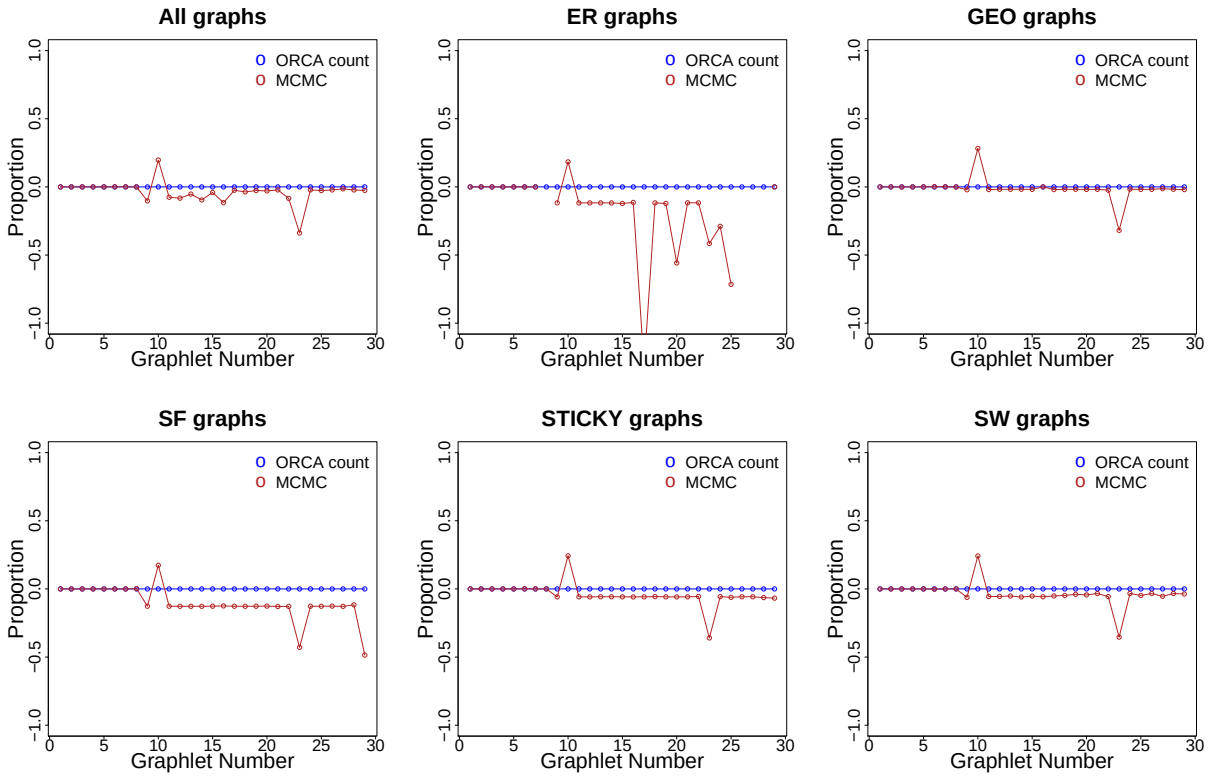


Figure 4.3: The figures above represent the log ratio between sampled graphlet counts from the MCMC method and the graphlet counts of ORCA across all synthetically generated networks of different types. The sampled proportions above were obtained by the MCMC sampling algorithm of Chen et al. [2016].

that the graphlets which are highly correlated in SF do not have the same correlations in GEO and vice versa. Despite the bias that occurs in both methods, different network types result in differently patterned correlation matrices. This is clearly seen in the negatively correlated (red area) of the NBE corrplots, whereas the corresponding regions of the MCMC corrplots are more loosely correlated (pale/white areas). A similar trend occurs in some of the real world graphs as shown in the GCM visualizations in Figure 4.5. The full-count GCM visualizations are also different from each other because the correlations are different in SF and GEO networks. However, the sampled GCMs look completely different from the full-count GCM. More work needs to be done to fully comprehend why the correlations are so different, apart from the fact that they are sampled.

Since GCMs are created from graphlet counts, we could not use the graphlet concentrations (as in the previous section) but instead used graphlet orbit signature matrix functionality of BLANT. This means that even the MCMC algorithm would not produce unbiased samples because the signature matrix requires raw counts. We would expect NBE to sample a few graphlets from many different locations in the network, whereas MCMC would have sampled many graphlets from a few randomly selected regions. This accounts for the main differences seen in the GCM visualizations.

4.6.3 Multi-Dimensional Scaling of Networks from NBE Sampling

Multi-Dimensional Scaling on Sparse Synthetic Networks Using NBE

Figures 4.6 and 4.7 show each graph from our Sparse Synthetic Network Database plotted as a point in space after MDS, for various values of k and m. The GCM matrices used to do the MDS were acquired from the NBE graphlet sampling method. As both the size of graphlets k, and the number of samples m increase, the disparity between network models becomes even more apparent. This is evidenced by the increase in inter-cluster distances for $k = 3$

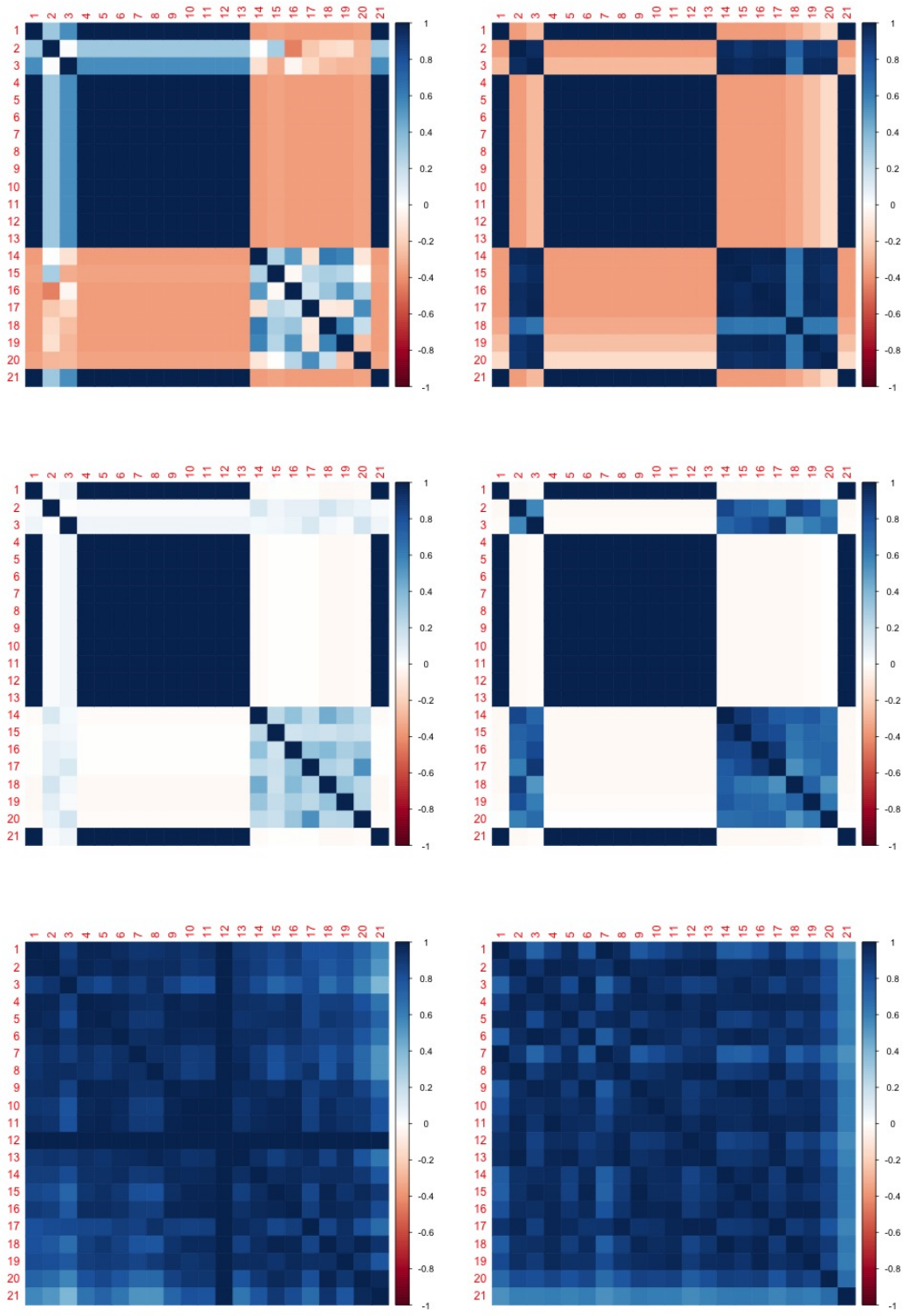


Figure 4.4: Top Row: Corrplots of GCM made from NBE graphlet sampling. Middle Row: Corrplots of GCM made from MCMC sampling. Bottom Row: Corrplots of GCM made from full counts of ORCA. Left Column: a Geometric network of 6000 nodes and density 0.0075. Right Column: a Scale Free network of 6000 nodes and density 0.0075.

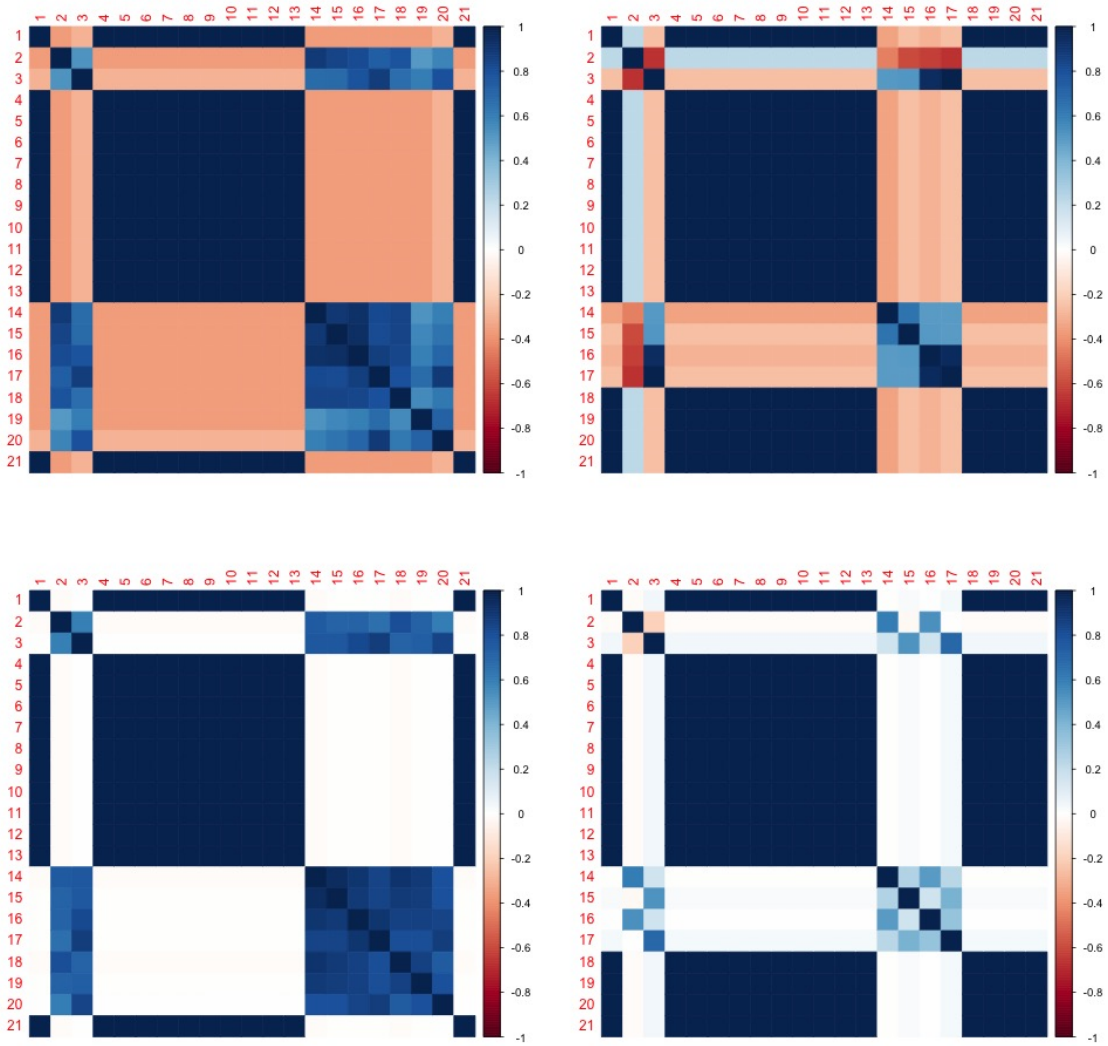


Figure 4.5: Left: a Facebook network - sofb-UCF52 of 14939 nodes and density 0.00384. Right: a Gene-μRNA network - Brassica_napus of 21076 nodes and density 0.00184. Top Row: Corrplots of GCM made from NBE sampling. Bottom Row: Corrplots of GCM made from MCMC sampling. The matrices highlight different correlations between different pairs of graphettes and these correlation trends vary with network model.

and 10^6 samples where the axes range within $(-1.5, 1.5)$ (Figure 4.6) whereas for $k = 7$ and 10^6 samples, the x-axis ranges over a vastly larger range of $(-2000, 2000)$ (Figure 4.7). We also begin to observe intra-cluster groupings which may be representative of small variations such as density, size or other network model attributes or the way the graphs were created. Figure 4.6 also shows the clusters formed from an ORCA count on the same sparse synthetic networks, for up to $k = 5$, and also contains some intra-cluster disparities, as expected from Yaveroğlu et al. [2014]. It must be emphasized that ORCA produces graphlet counts for all values of k up to and including the one specified (in this case, 5). Therefore, it provides much more information about local connectivity of nodes. BLANT samples those graphlets with *exactly* k nodes.

4.6.4 Multi-Dimensional Scaling on Dense Synthetic Networks Using NBE

In spite of the networks in this dataset having starkly different sizes and densities, Figure 4.7 shows unmistakable indication of inherent groups within the graphs of the dense synthetic network database. Though the difference between clusters is not as unambiguous as for the sparse synthetic networks, there is enough clarity to conclude sampling 10^7 7-graphlets begins to separate the graphs of different network models. Further, perhaps by sampling graphlets of a larger size or by simply taking more samples, we would obtain more distinct clustering. Graphs in this database have an average node degree of 2540 with a maximum average degree of 9375. A full graphlet count on one of these networks using ORCA would have taken a few weeks or even months. BLANT, on the other hand, took around $25 - 45$ minutes to take 10^7 samples from a single graph in this network. BLANT can therefore be of great assistance in drawing conclusions about the topological nature of a graph in a tremendously quicker time frame.

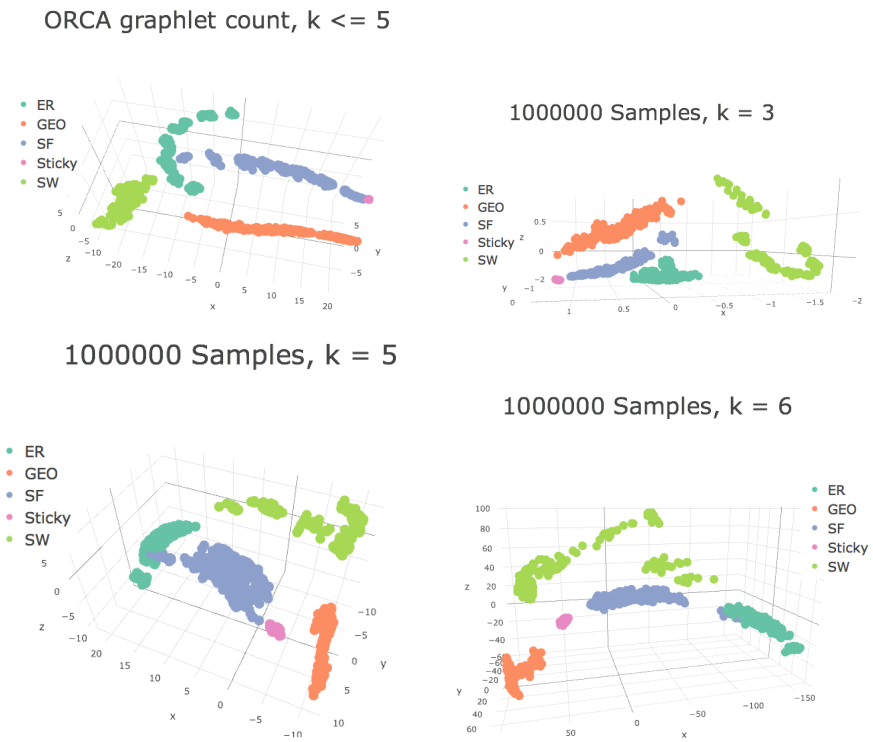


Figure 4.6: Left to Right: MDS of sparse synthetic database networks using distances computed from ORCA's exhaustive enumeration of graphlets for $k \leq 5$. After taking 10^6 samples of 3-graphlets and computing pairwise graphlet correlation distances, each network from the sparse synthetic database was scaled into 3 dimensions by MDS and plotted as shown. The right two images show MDS on sparse synthetic networks using distances computed from 10^5 samples of 5-graphlets (left) and 10^3 samples of 6-graphlets. These graphlets were sampled using NBE.

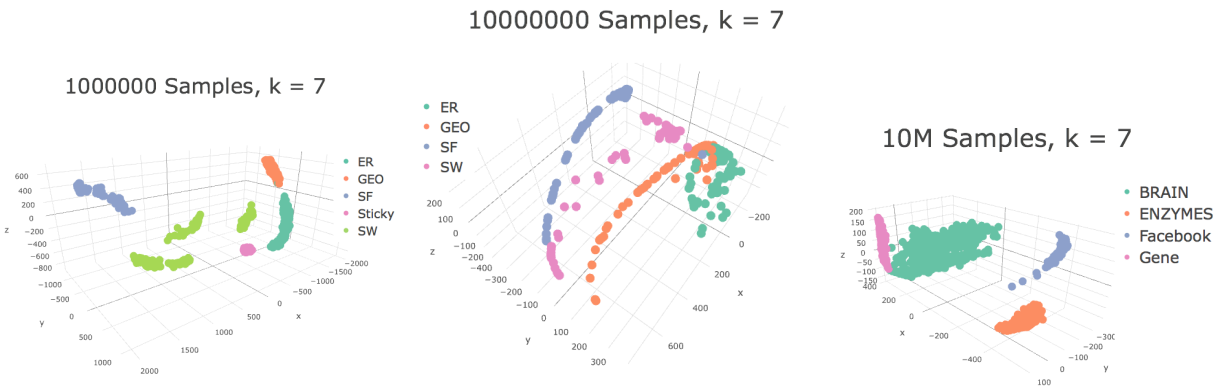


Figure 4.7: Left: MDS of sparse synthetic networks using distances computed from 10^6 samples of 7-graphlets. Middle: MDS of dense synthetic networks using distances computed from 10^7 samples of 7-graphlets. Right: MDS of real world networks using distances computed from 10^7 samples of 7-graphlets. The graphlets were sampled using NBE.

Multi-Dimensional Scaling on Real World Networks Using NBE

Figure 4.7 shows the separation in Real World Networks. We see clear separation between four different types of networks after taking 10 million samples of 7-graphlets using NBE.

4.6.5 Multi-Dimensional Scaling of Networks Using MCMC Sampling

Multi-Dimensional Scaling on Sparse Synthetic Networks Using MCMC

Just as for NBE, we observe that the inter-cluster separation increases as k increases when taking samples using MCMC. In addition, we begin to see patterns within clusters (Figure 4.8). The SW cluster breaks into three parts and though we see this behaviour as well with NBE, the reason is not clear in NBE. With the MCMC clusters, each SW cluster is comprised mostly of networks of a different density. Similarly, the ER cluster becomes more elongated as k increases and we observe here that the density of networks increases from one end of the cluster to the other, whereas this trend is not clear from NBE clusters, despite their similarity to MCMC clusters.

We also observe that the relative position of clusters in the MCMC MDS when compared with the relative positions of NBE MDS in Figure 4.6. This may be due to the bias in both sampling algorithms. In particular, if we use the ORCA clusters in Figure 4.6 as a gold standard, we notice that the Sticky cluster is very closely positioned next to the SF cluster and somwhat close to the GEO cluster. We observe that this positioning is consistent up to $k = 5$ under both NBE (Figure 4.6) and MCMC (Figure 4.8). However, for $k > 5$, the position of the Sticky cluster shifts closer to ER and partially to SW when NBE sampling is used, but remains close to SF and GEO clusters for $k > 5$ when MCMC sampling is used.

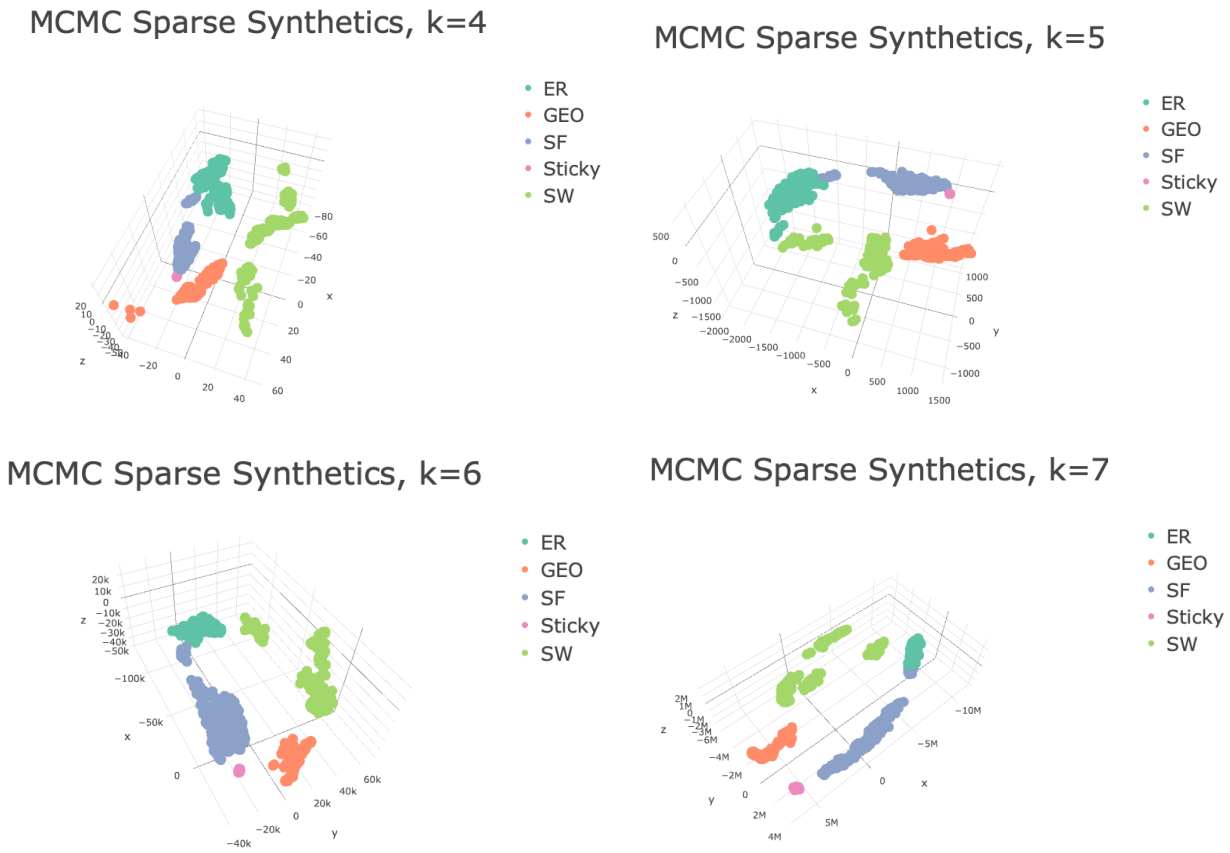


Figure 4.8: After taking 100000 samples of $4, 5, 6, 7$-graphlets and computing pairwise graphlet correlation distances, each network from the sparse synthetic database was scaled into 3 dimensions by MDS and plotted as shown. The graphlets were sampled using the MCMC method.

Multi-Dimensional Scaling on Dense Synthetic Networks Using MCMC

MDS on the GCD distances acquired from MCMC sampling is also evidence that sampling is an effective and efficient way to separate networks that are huge. With 1 million samples for $k = 6$, we are able to see separation on the dense synthetic networks (Figure 4.9). The long arcs that we observe for GEO, SW and ER networks are comprised of networks of increasing density (densities ranging from 0.1 to 0.98).

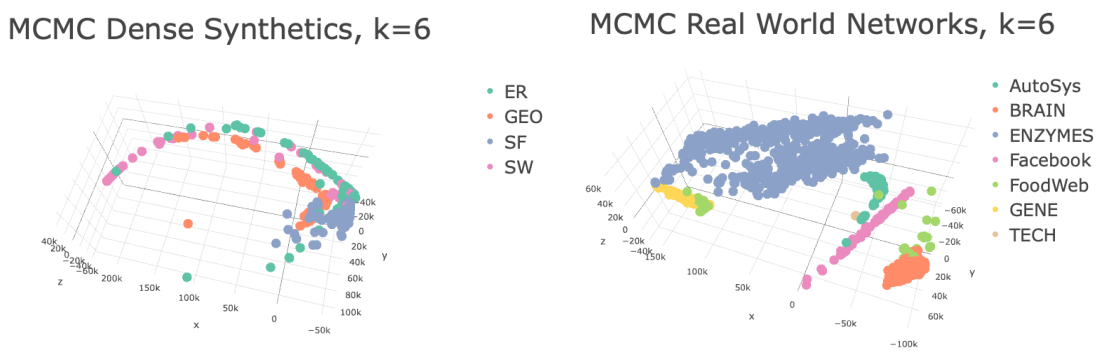


Figure 4.9: After taking 1 million samples of 6-graphlets and computing pairwise graphlet correlation distances, each network from the dense synthetic database and from the real-world network database was scaled into 3-dimensions using MDS as shown in the images above. The graphlets were sampled using the MCMC sampling method.

Multi-Dimensional Scaling on Real World Networks Using MCMC

We also performed MDS using the GCD pairwise distances between real-world networks of various types. We sampled 1 million k-graphlets using MCMC sampling. We were able to separate more types of networks with fewer k-graphlet samples and using smaller value of k, than when NBE sampling was used. The FoodWeb and Gene-μRNA networks cluster very closely together because they are both bi-partite graphs and hence, are structurally quite similar. The networks shown are quite well-separated (Figure 4.9).

4.7 Run Time from Sampling

Table 4.2 shows the run times of BLANT on different biological networks from BioGRID, Collins and Network Repository. The number of samples taken per second for EBE, NBE, MCMC are shown. All the sampling methods output hundreds of thousands of samples per second, except for EBE which outputs millions of samples per second. We can achieve greater speedups by using the multi-threaded option. Though BLANT's speed is far superior to that of ORCA or JESSE, the take-away is that we need only a sample of graphlets to understand the topology of the network, instead of waiting hours, or days to arrive at a similar understanding with a more thorough graphlet count.

4.8 Conclusion and Discussion

We have experimented with different graphlet sampling methods and showed that NRE extracts a less biased sample of graphlets than EBE and NBE, but still has a high bias. The MCMC method is also biased, but adjusts for this bias by dividing each graphlet count by an overcount factor and hence, produces the best estimated concentration of graphlets. We have implemented all of these methods into our tool BLANT, used for sampling graphlets, creating a database of indices, counting graphlets and creating graphlet orbit signature matrices. BLANT samples tens of thousands of graphlets in just a second, much faster than any exhaustive counting technique. BLANT sampling together with the Graphlet Correlation Measure is enough to cluster together networks of the same model and separate networks of different models, in a similar way to the clustering produced after exhaustively enumerating graphlets as in Yaveroğlu et al. [2014]. However, the relative distances of networks change according to which sampling method is used as we have seen in Figures **??** and 4.8. This means that it is important to be able to either sample unbiasedly to begin with, or to

Table 4.2: **Performance**: BLANT's sampling rate (unit = **thousands** of graphlets per second, single-core) of various networks, including some huge BNU Brain Networks from the Network Repository; r_k=sampling rate, per core, in thousands of graphlets per second, for graphlets of size k; $\overline{deg}$=mean degree. RAM usage was typically 1-2GB and averaged just over 3GB for the BNU networks. All experiments performed on an 8-core 3.5GHz Intel Xeon E5-1620 v3 CPU with a 10MB CPU cache and 32GB of RAM.

Source	Network	nodes	edges	$\overline{deg}$	method	r_3	r_4	r_5	r_6	r_7	r_8
Collins	syeast-1k	1004	8323	16.6							
					MCMC	1206	713	487	357	251	91
					NBE	895	615	443	406	335	271
					EBE	2222	1452	1024	834	677	542
BioGRID	RNorvegicus	1657	2330	2.81							
					MCMC	1919	1795	1277		698	253
					NBE	608	434	358	284	296	263
					EBE	2382	1696	1287	1030	840	710
	CElegans	3134	5428	3.46							
					MCMC	1782	1400	932	725	532	510
					NBE	744	601	644	591	493	367
					EBE	1865	1467	1193	920	747	581
	AThaliana	5897	13381	4.54							
					MCMC	1838	1231	932	672	468	211
					NBE	656	658	550	496	422	333
					EBE	1782	1438	1084	867	695	576
	DMelanogaster	7937	34753	8.75							
					MCMC	1438	1015	760	566	429	281
					NBE	765	544	443	366	313	270
					EBE	1862	1310	970	777	622	479
	SCerevisiae	5831	77149	26.5							
					MCMC	1209	940	685	504	390	88
					NBE	349	225	188	170	147	124
					EBE	1840	1189	939	732	580	358
	HSapiens	13276	110528	16.7							
					MCMC	1173	896	643	463	337	106
					NBE	393	293	212	180	155	130
					EBE	1662	1179	889	692	554	348
BNU	0025878	699,697	127,906,128	365.6							
					MCMC	7.1	6.5	6.2	5.7	5.2	4.9
					NBE	16.6	13.4	11.5	9.82	8.62	6.51
					EBE	55.8	48.4	41.5	35.2	29.5	21.7

accurately estimate the number or graphlets and/or orbits.

Chapter 5

Network Generators Do Not Preserve Graphlet Distributions And Other Properties

5.1 Synthetic Network Generators

Graphs are used to represent several real-world relationships such as: people (nodes) and their interpersonal relationships (edges may represent business, friendship, family relationships etc), internet network routers (nodes) and the links (edges) between them, protein interactions, gene regulations, library dependencies within a programming language, predator-prey relationships, etc. Analyzing network representations of real-world phenomena can provide insights into evolution, function and dynamics of the system Newman [2010]. Studying the structure and topology of networks can enable a higher level of understanding of the system being modeled and this brings about a need for synthetic networks which replicate the structure and properties of the real-world phenomenon. We may need to generate graphs for

extrapolations, predictions, simulations and hypothesis testing in cases where it is difficult to curate real data, or where there are security risks or confidentiality clauses preventing the release of real data to the public.

In the process of designing network generators, we must think about which properties of the real-world network should be passed on to the synthetic network. Which properties are the most characteristic of the real-world network? How should we measure the similarity (or distance) between two networks? Network generation is quite a challenging problem and there has been much prior work on generating synthetic graphs to match real-world networks, described in surveys in Chakrabarti and Faloutsos [2006], Brown [2009], Dunlavy et al. [2009]. There are two main categories of network generation methods: *generative* models and *editing* models.

Generative models usually begin with a small initial seed graph and produces a graph via randomization and replication, which matches some pre-specified properties of the original graph. These pre-specified properties may include some combination of degree distribution, clustering coefficient, diameter, and other standard network properties. Early models include ER and ERDD models, which attempt to fit the degree distribution of the original network into the synthetic network but are unable to reproduce other important characteristics such as power-law distribution (if present in the original network), eigenvalues, etc. In Medina et al. [2000], an attempt is made to preserve the power-law property of networks other standard network properties are not matched. RMAT was developed to match several properties of networks including degree distribution, 'small diameter' property of real-world networks, the community-like structure of real-world networks, k-hop distribution, etc. Chakrabarti et al. [2004]. RMAT recursively subdivides the adjacency matrix into 4 sections, and each edge is dropped into partition i using probabilities a_1, a_2, a_3, a_4 such that $\sum_{i=1}^{4} a_i = 1$. One generative model which has gained popularity over the past decade Gutfraind et al. [2015] is Kronecker (and stochastic Kronecker graphs) Leskovec et al. [2010], which also matches

many of the standard network properties namely degree distribution, diameter, K-hop distribution, and scree plots of eigenvalue vs rank. Kronecker graphs begin with an initiator graph and recursively grows the graph using the Kronecker product.

The other type of network generator, graph editing, begins with the original network and perturbs the graph in some way to introduce sufficient variability while preserving its overall structure Mihail and Zegura [2003]. Musketeer is one such graph editing model Gutfraind et al. [2015]. Beginning with the original network, Musketeer creates a hierarchy of graphs, each one inheriting properties from the ones before it. Local changes are made to each network in the hierarchy, affected only the local topology and the edit rate is gradually lowered over the hierarchy at which point, the replicas are expected to be arbitrarily close to the original network.

BTER is another graph generator which does not use an iterative process to make the synthetic network but creates the graph directly from the input degree distribution and clustering coefficient distribution Kolda et al. [2014]. It begins by assigning each node a degree based on the input degree distribution. Nodes are grouped into blocks and edges are added into each block so that the nodes achieve thir input clustering coefficient while not exceeding their degree. Finally, the blocks are joined together to form the network. As discussed previously, STICKY can also be considered a network generator because it attempts to replicate the specific input network.

Though network generators are often created to match specific network properties, none have been designed to model the graphlet distribution of the input network. We briefly show that several state-of-the-art network generators are not able fundamental properties such as degree distribution, diameter, clustering coefficient and K-hop distribution. In addition, we show that they do not preserve the graphlet distribution on most types of networks.

We examine the performance of several popular state-of-the-art network generators: BTER,

RMAT, MUSKETEER, KRONECKER and STICKY, on different types of real-world networks. We measure how each of the generators preserve well-known graph properties: diameter, average clustering coefficient, and degree distribution. We also test their ability to preserve k-graphlet distribution of the original real-world networks. We find that graphlet distribution is not preserved in most types of networks, that traditional network properties are also not always preserved and, using the GCD network measure, the synthetic networks generated by each generator are topologically different across generators.

5.1.1 Properties of Networks

Definition 6. *The **degree distribution** is the probability distribution of the degrees of all nodes in graph $G(n, m)$. The distribution may be theoretical or empirical.*

Definition 7. *The **global clustering coefficient** C of $G(n, m)$ is proportional to the number of triangles in G and is defined as $C = 3 * \frac{number\ of\ 3\text{-}graphlets}{total\ number\ of\ triplets}$.*

Definition 8. *Consider a node v in G and its neighbors $Neigh(v)$. Let S be the set of edges between the nodes of $Neigh(v)$. More formally, $S = \{(u, w)|u \in Neigh(v), w \in Neigh(v), u \neq w \neq v\}$. The **local clustering coefficient** of node v or **clustering coefficient** of v denoted as C_v, is the ratio of number of edges between the nodes of $Neigh(v)$ to the number of edges that could theoretically exist between them, i.e. $C_v = \frac{2|S|}{|Neigh(v)|*(|Neigh(v)|-1)}$.*

Definition 9. *The **diameter** of a network $G(n, m)$ is the length of the longest, shortest path in between any two nodes in G.*

5.2 Method

For each synthetic network generator (RMAT, BTER, STICKY, Kronecker and Musketeer), we created one synthetic version for each of many 7 Technology, 98 Facebook, 733 Au-

tonomous Systems and,535 GENE-μRNA networks. We then observed network properties (degree distributions, diameters, clustering coefficients, and K-hop distributions) of each generator's synthetically generated network and compared with the original real-world network. We also used MCMC sampling to compare the log ratio graphlet distributions of the synthetically generated networks with the corresponding real-world network.

5.3 Results

Figure 5.1 shows the degree distribution of 4 TECH networks. Though the generators follow the same trend, they are different from the original degree distribution.

Figure 5.2 shows the clustering coefficient and diameter of 7 TECH networks and 48 Retweet networks. All the synthetic networks are within $5-6$ of the true diameter, but they rarely ever match the diameter. Similarly, the clustering coefficients are on the order of 10^{-1} away from the true clustering coefficient. Since clustering coefficients range between 0 and 1, this is actually a large deviation.

Figure 5.3 shows the log of the graphlet distribution for $k = 3, 4$ and, 5 for a TECH, Facebook, Autonomous System and GENE-μRNA network. The graphlet distribution matched well with the original on the Facebook graph but not on the others.

For each of the synthetic networks of a given real-world graph, we computed $1-GK$ between every pair (including the original network) for various values of k, to get their pairwise distance, and used an MDS to project each network's relative position from each other in one dimension. For a given network type, we show in Figures 5.4, 5.5, 5.6 and, 5.7 the real-world networks relative to their corresponding synthetics in increasing order of coordinate positioning together with the positions of the synthetic networks. There is no relationship between *any* synthetic and its corresponding real-world network. This shows not only that the syn-

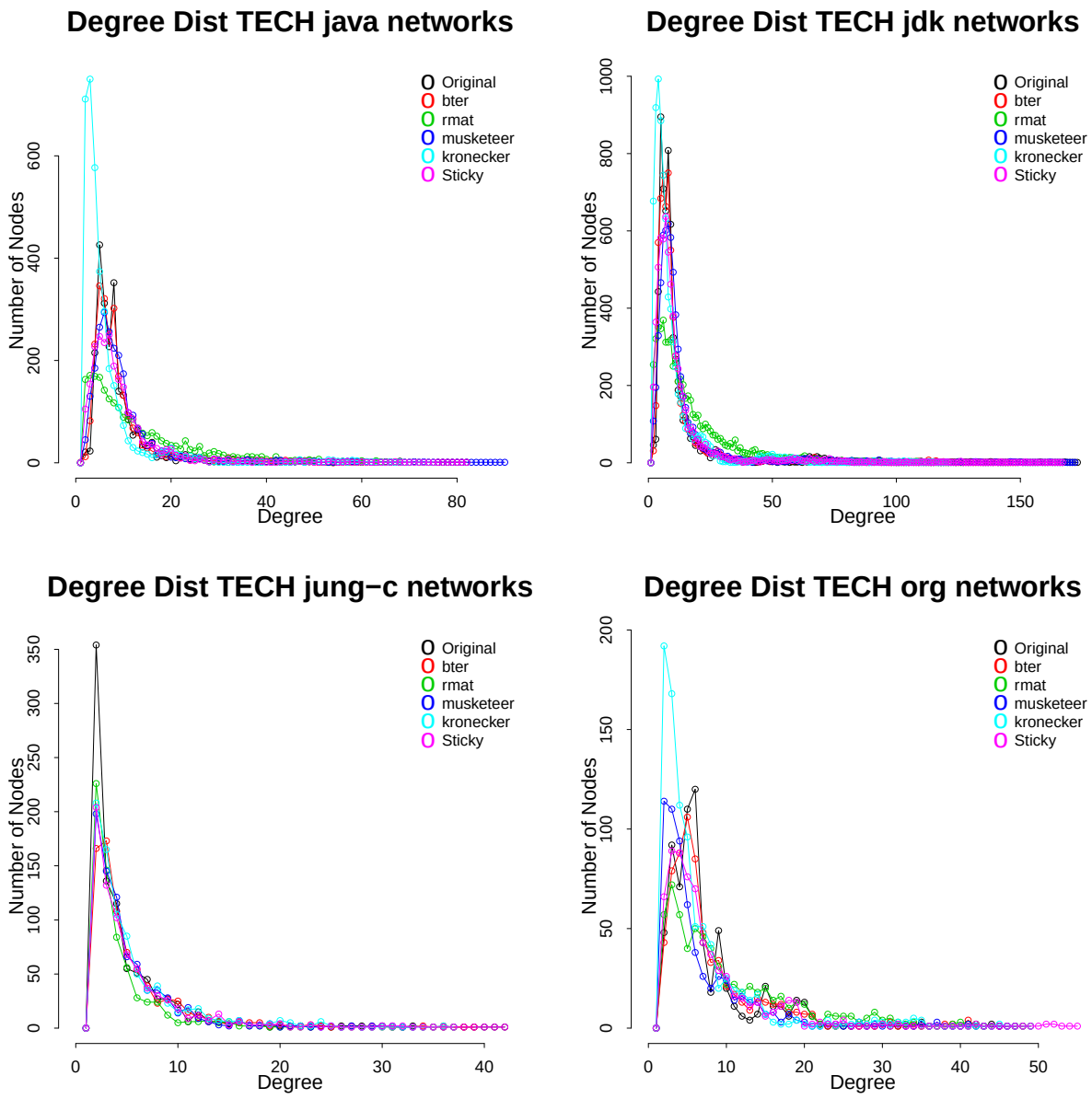


Figure 5.1: The degree distribution of various TECH networks. Each point represents the number of nodes (y-axis) having a specific degree (x-axis). The original network's degree distribution is shown in black and is markedly different from the synthetics.

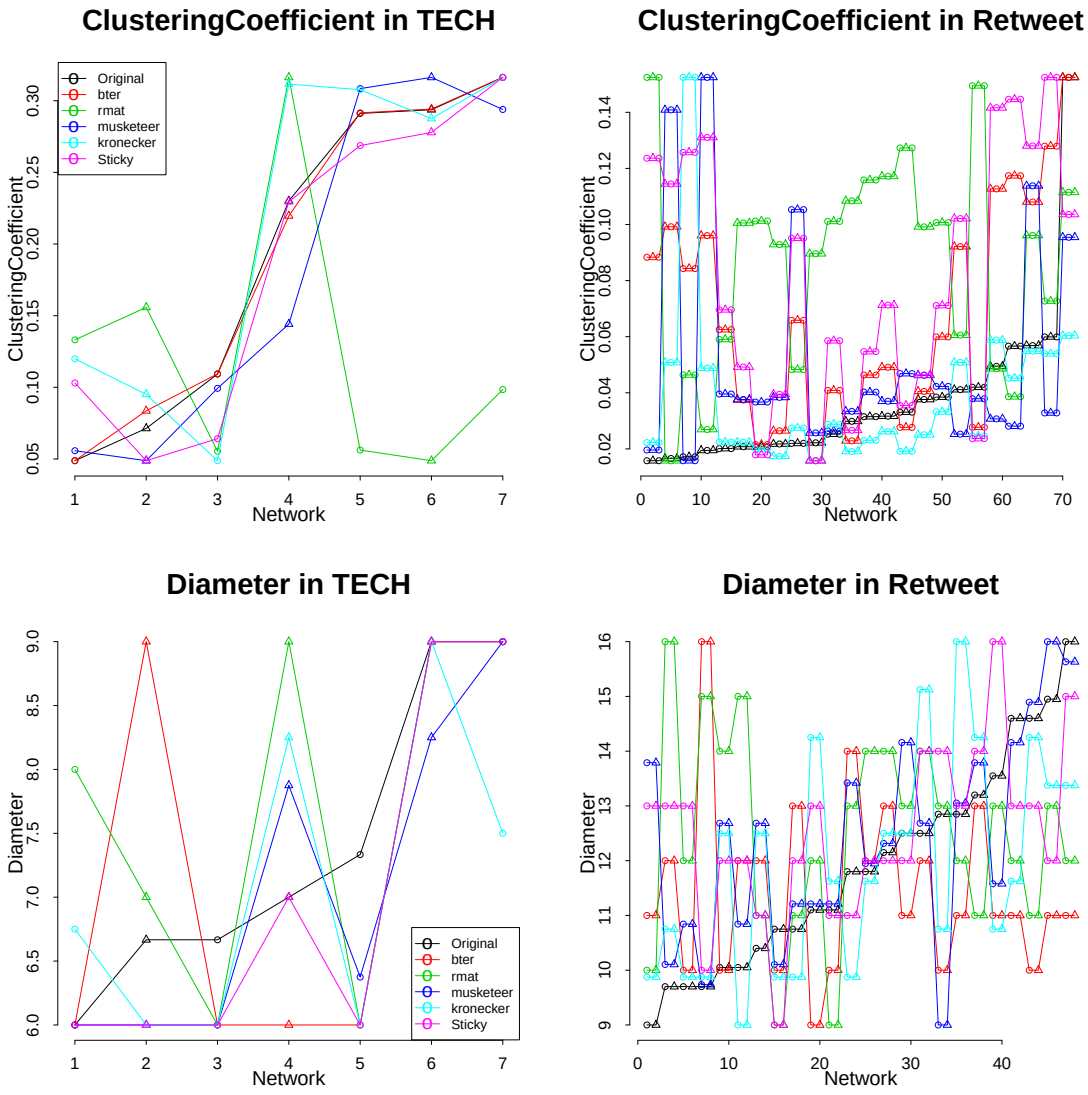


Figure 5.2: Global Clustering Coefficient and Diameter for the synthetic networks from all the generators on 7 different TECH networks and 48 Retweet networks.

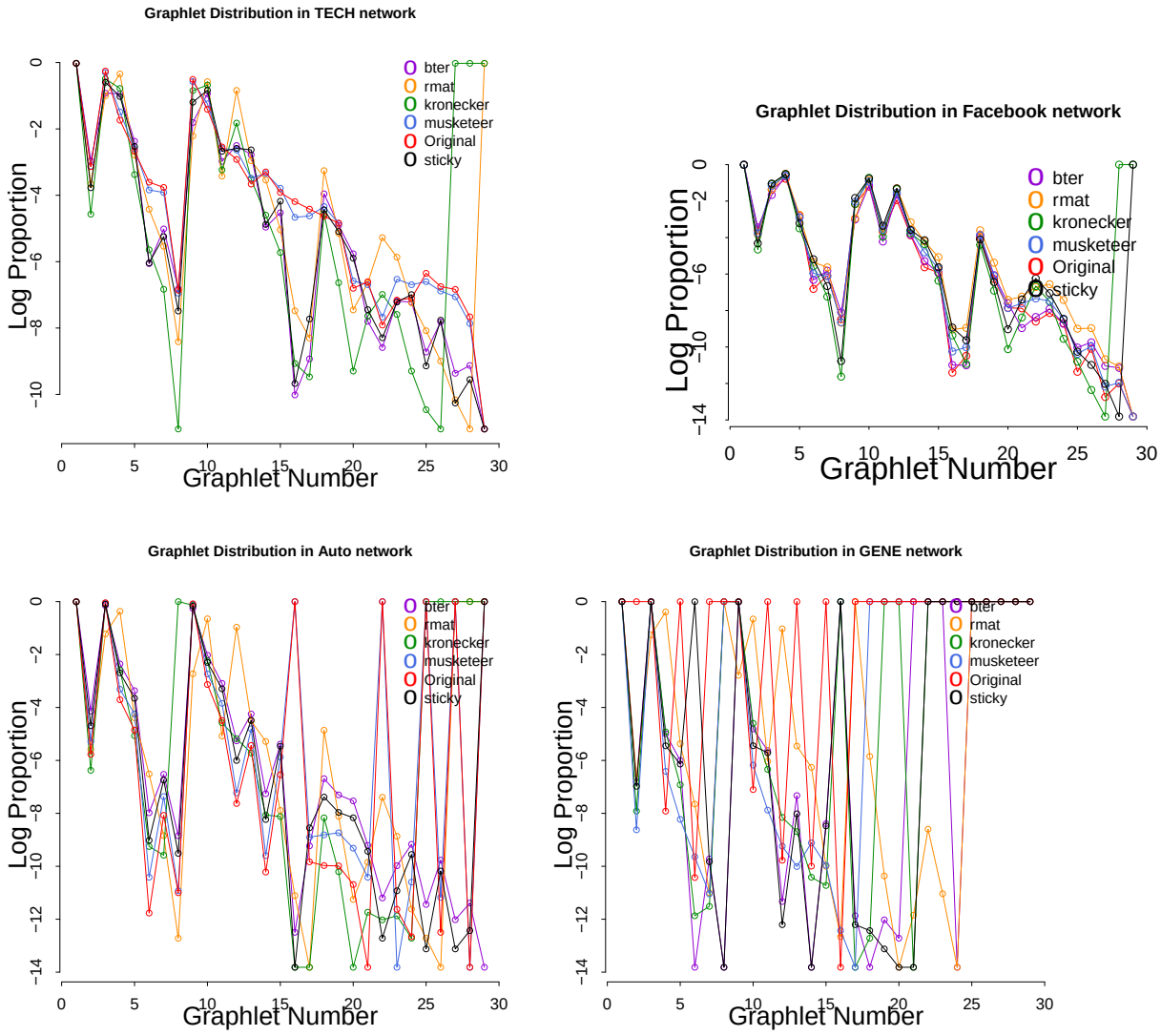


Figure 5.3: Log of the Graphlet Distribution from the synthetically generated networks compared with the original log distribution for $k = 3, 4$ and, 5 on various types of networks.

thetic networks are all different from each other in structure, but also that the generators themselves are not sensitive enough to the graphlet structure.

While synthetic versions of real-world networks should have some variability within them due to randomness, we expect that they would have smaller deviations that that shown above. In fact, K-S tests on the degree distributions of the real-world networks of TECH and Retweet networks and their associated synthetics from each generator all returned low p-values (below 10^{-4}), indicating that the two distributions were different.

5.4 Summary and Future Work

The current state-of-the-art network generators do not preserve the graphlet topology of a real network in their synthetic networks. In addition, they do not always conserve other network properties such as diameter and degree distribution. Though network generation is a difficult problem, there is need for a network generator that can preserve all of these properties.

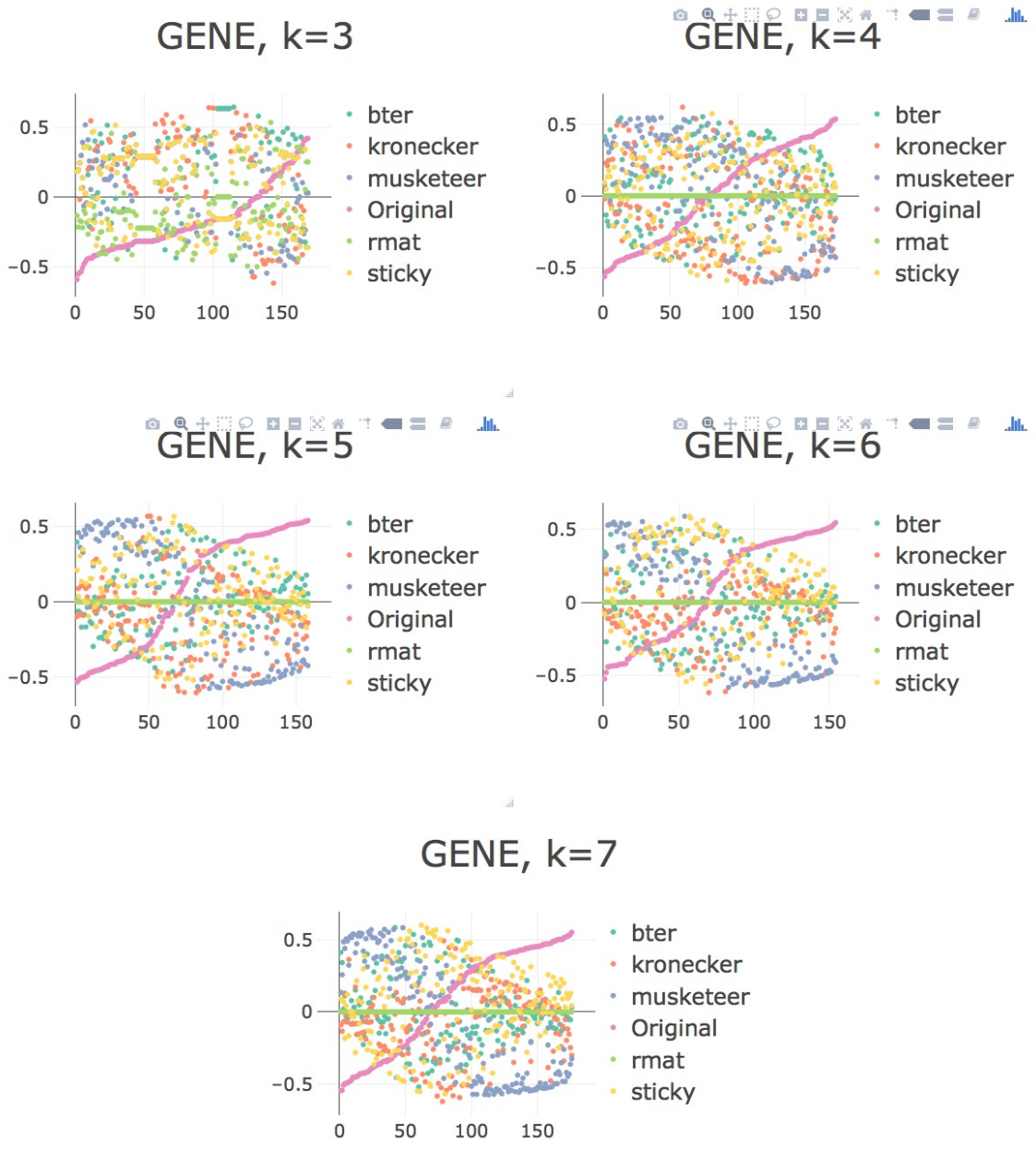


Figure 5.4: 1-dimensional MDS on $1-GK$ scores between every pair of Gene-μRNA networks (original and synthetic), using graphlet concentrations for up to 7-graphlets. The graphs are ordered according to the relative positions of the original networks on the vertical axis.

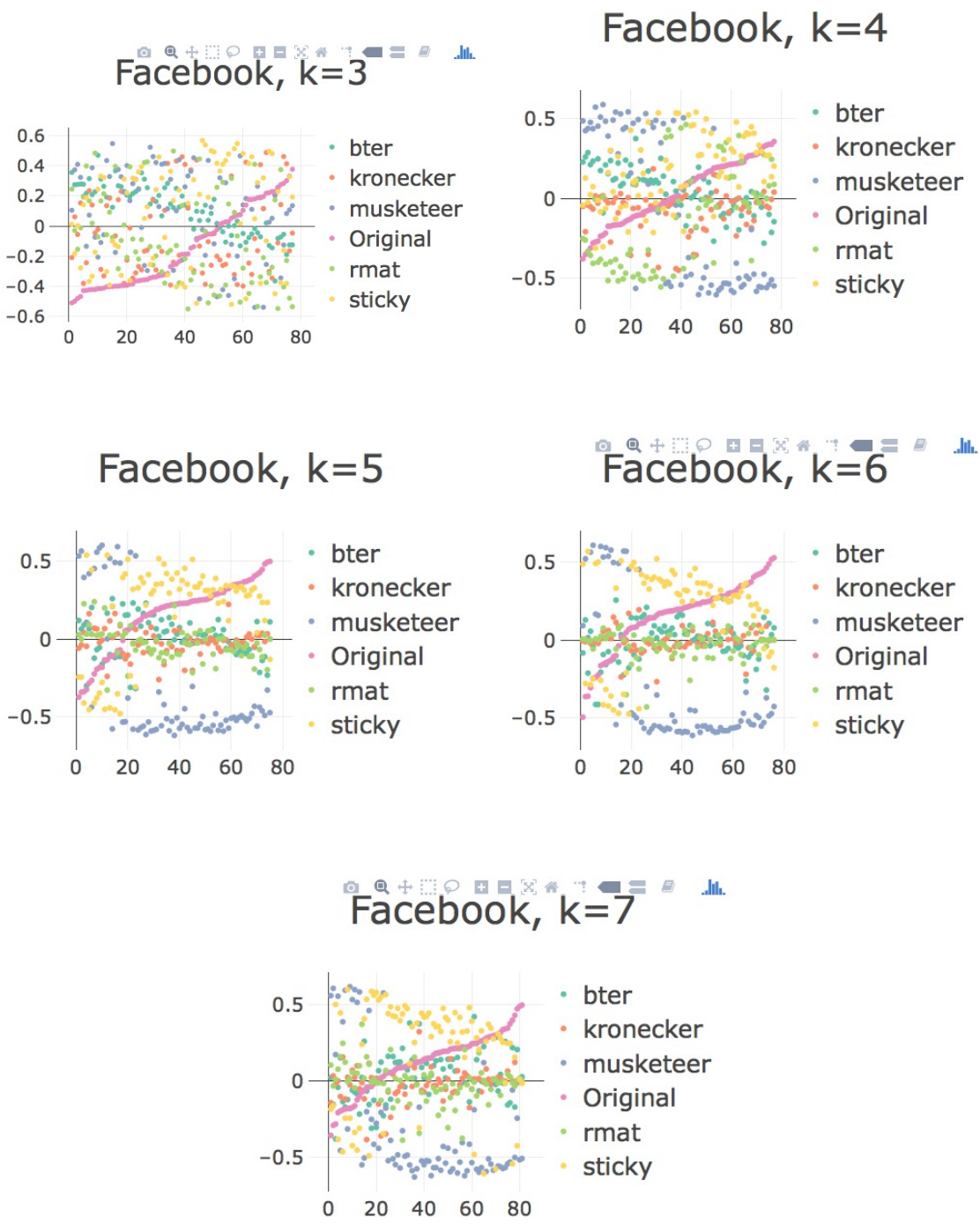


Figure 5.5: 1-dimensional MDS on $1 - GK$ scores between every pair of Facebook networks (original and synthetic), using graphlet concentrations up to 7-graphlets. The graphs are ordered according to the relative positions of the original networks on the vertical axis.

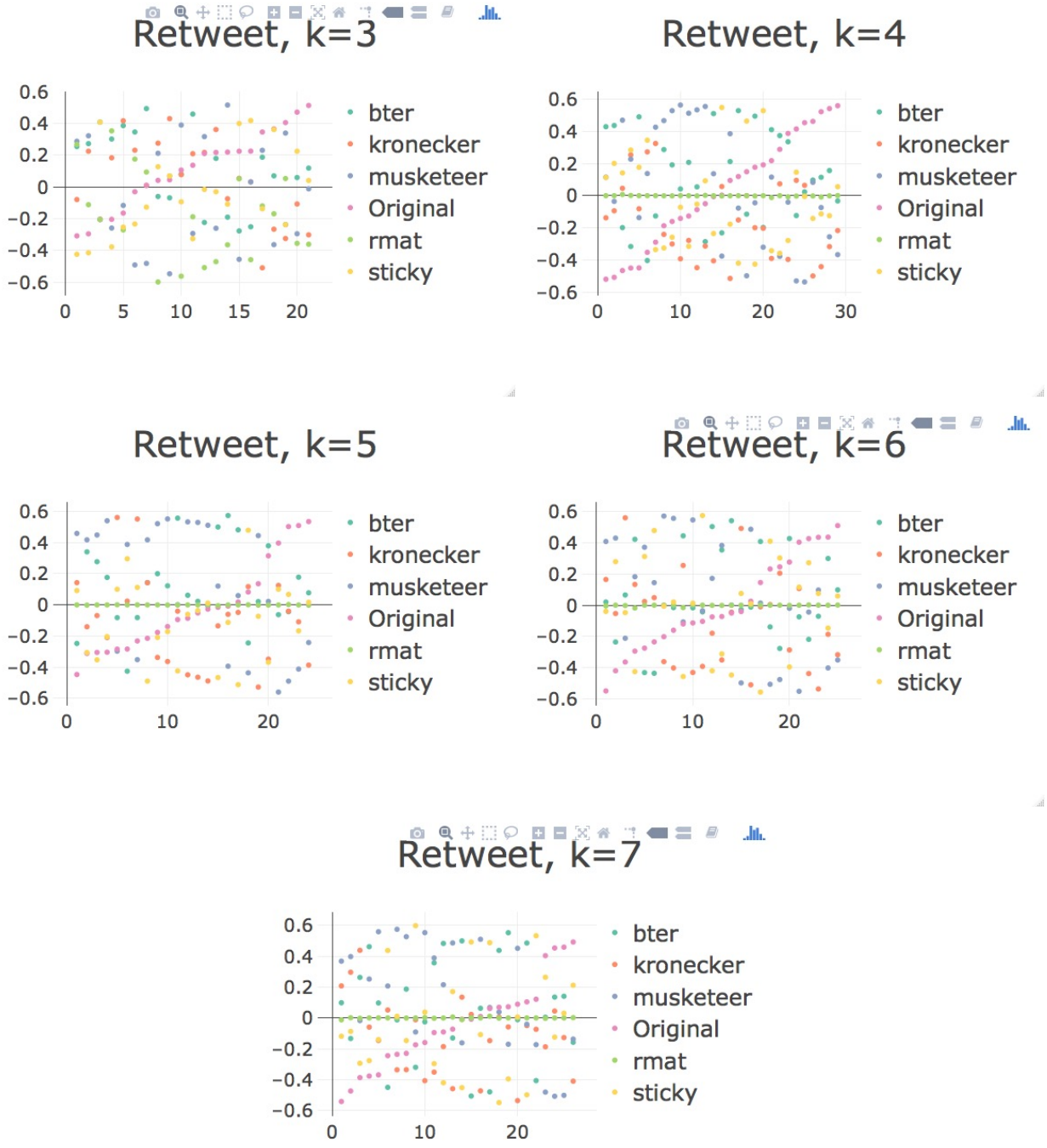


Figure 5.6: 1-dimensional MDS on $1 - GK$ scores between every pair of Retweet networks (original and synthetic), using graphlet concentrations up to 7-graphlets. The graphs are ordered according to the relative positions of the original networks on the vertical axis.

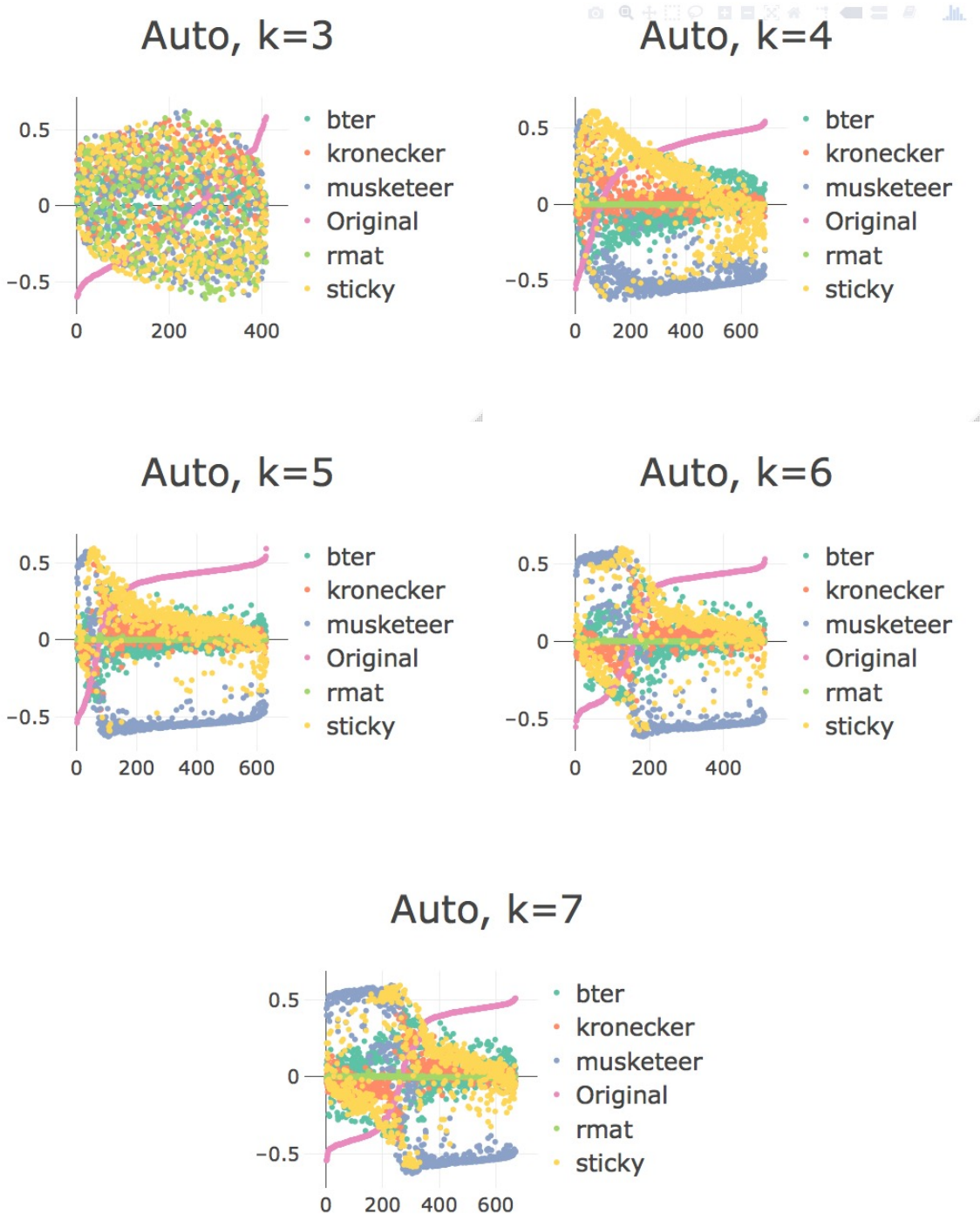


Figure 5.7: 1-dimensional MDS on $1-GK$ scores between every pair of Autonomous Systems networks (original and synthetic), using graphlet concentrations up to 7-graphlets. The graphs are ordered according to the relative positions of the original networks on the vertical axis.

Chapter 6

Brain Networks in OCD Patients

Obsessive-Compulsive Disorder (OCD) is a psychiatric condition characterized by recurrent, intrusive, disturbing thoughts or actions [APA, 1994]. OCD negatively affects social functioning [Piacentini et al., 2007] and quality of life [Lack et al., 2009] and is associated with hyperactivity in caudate and putamen, thalamus, anterior cingulate, and orbitofrontal cortex [Maia et al., 2008, Menzies et al., 2008]. The disorder is prevalent in $2.5 - 2.7\%$ of children and adolescents [Rapoport et al., 2000, Heyman et al., 2003]. Though some treatments are successful, they sometimes have no response and their effects can be unpredictable [Knopp et al., 2013]. There have been several studies attempting to understand the connectivity differences between healthy individuals and patients with OCD [Harrison et al., 2009, Meunier et al., 2012, Schlösser et al., 2010] but it has become necessary to search for deeper insights into OCD patho-physiology by searching for dysfunction in global and local systems [Anticevic et al., 2014, Fitzgerald et al., 2010, Milad and Rauch, 2012, Stern et al., 2012, Zhang et al., 2011].

A few attempts have been made to study the brain connectivity patterns of patients with Obsessive-Compulsive Disorder (OCD) using graph theory analysis. The network is typi-

cally constructed from resting state functional magnetic resonance imaging (rsfMRI) data, where nodes are voxels (i.e. the 3D analogy of a pixel) representing a region of the brain and edges represent a high correlation in their activation during rsfMRI. It was found that adult OCD patients showed decreased functional connectivity (compared with healthy adult controls) in the posterior temporal regions and increased connectivity in various control regions such as cingulate, thalamus and cerebellum [Zhang et al., 2011]. In addition, the network of healthy patients had high clustering coefficients and short path lengths. This means, healthy patients exhibited small-world architecture. However, OCD patients showed significantly higher local clustering, implying abnormal functional organization in the OCD brain network [Zhang et al., 2011]. [Shin et al., 2014] studied the effect of 16 weeks of pharmacological treatment on drug-free adult OCD patients. Prior to treatment, it was observed that the OCD patients exhibited decreased signs of small-world connectivity (i.e. lower local clustering coefficients, local efficiency), decreased functional association between default-mode and frontoparietal modules and altered connectivity degrees in many brain areas overall. After treatment, there was a statistically significant increase in small-world efficiency, modular organization and connectivity degree, especially in the right ventral frontal cortex in OCD patients [Shin et al., 2014]. Another study found that OCD patients showed increased functional connectivity primarily within the CSTC circuits and decreased functional connectivity in the occipital cortex, temporal cortex and cerebellum [Hou et al., 2014]. One study on OCD children patients found that small-worldness and modularity were lower in OCD patients than in healthy controls [Armstrong et al., 2016]. They also found that in the frontopolar, supplementary motor, sensorimotor and cortices with lower betweenness centrality, the local clustering coefficients were higher in OCD patients. These findings are consistent with more locally intensive connectivity or less interaction with other brain regions at these sites. Whole-brain data-driven graph theoretical analysis in Vaghi et al. [2017] disclosed that striatal regions constitute a cohesive module of the community structure of the functional connectome in OCD adult patients as nodes within the basal ganglia and cerebellum were

more strongly connected to one another than in healthy control subjects.

Most of these studies have only scratched the surface of graph theory. Network analysis tools have the potential to unearth new paradigms to the structure, function and evolution of the real-world phenomenon being modeled. In this study, we take a closer look at the brain connectivity within 19 OCD children patients compared with 17 healthy children as a control group (CON) of similar age, using fMRI connectivity data from the study entitled 'OCD_and_control' downloaded from the USC Multimodal Connectivity Database Brown and Van Horn [2016]. We analyze the brain networks of OCD children patients and the CON children control group using both traditional graph properties as well as graphlet topology. While the concept of graphlets is not new, to the best of our knowledge, it has not yet been applied to the study of OCD brain networks. We introduce the notion of k-graphlet Edge Hamming Distance and k-graphlet Edge Hamming Distance Sequences and found that along some of the sequences there is a statistically significant linear trend in the ratio of graphlets in the OCD to CON networks. In addition, we study various network properties on both networks and find that the OCD brain connectivity is drastically different from the CON connectivity in terms of their node degree (i.e. co-activity in corresponding regions of the brain) and graphlet distributions. We also find that there are many more circuits in the OCD brain when compared with CON while there are many more clique regions in CON than in OCD. Further, the busiest nodes in each network are completely different from each other both in physical location and general function.

6.1 Method

6.1.1 Data and Networks

We used the two fMRI connectivity matrices from the study 'OCD and control' from the USC Multimodal Connectivity Database Brown and Van Horn [2016]. One matrix represented the average fMRI connectivity among 19 OCD children patients and the other represented the average fMRI connectivity among 17 healthy control children patients. We replaced any non-zero entries in the matrices with 1 to form binary matrices, which we used as the adjacency matrices for OCD and CON. While this may seem like a grossly simplifying assumption for weighted-edge networks, in this case all non-zero edge weights were in the range $1.3 - 1.7$ in CON and $1.5 - 1.7$ for OCD, and so we believe in this case binarization is a reasonable approximation to the data. In total, each network contained 200 nodes (parts of the brain) and 1990 edges (connectivity correlation from the fMRI). The data also included $3D$ coordinates of each node, representing their relative positions to each other in space.

6.2 k-Graphlet Edge Hamming Distances

The Hamming distance between two strings is the minimum number of character substitutions required to transform one of the strings into the other. We extend this idea to k-graphlets.

Definition 10. *The k-Graphlet Edge Hamming Distance between two k-graphlets g_1 and g_2 is the minimum number of edges which must be removed or added to g_1 in order to transform it into g_2 (up to isomorphism).*

'Up to isomorphism' means the edge addition and removal procedure is independent of how

the graphlet is drawn. When the context of k is clear, we abbreviate k-Graphlet Edge Hamming Distance to EHD. For example, consider graphlet g_1 which is a path of length 2 and closed-triangle graphlet g_2: $EHD(g_1, g_2) = 1$ because g_1 requires one additional edge to make it a closed triangle, as seen in Figure 6.1. Alternatively, we can say that graphlet 2 is Edge Hamming Distance 1 away from graphlet 1 because we would need to remove any 1 edge from 2 to transform it into 1. We will use the convention that the first parameter of EDH has fewer edges than the second. The k-Graphlet Edge Hamming Distance can be considered a local distance measure.

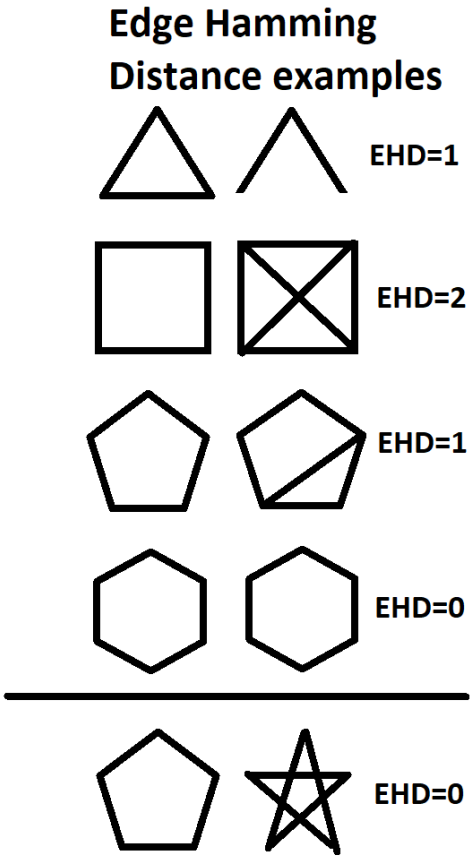


Figure 6.1: The Edge Hamming Distance is shown here for graphlets of various sizes.

6.2.1 Graph of k-Graphlets of Edge Hamming Distance-d

We can construct a directed graph H_d on all the graphlets of k-nodes in which each node is a k-graphlet and an edge exists between two nodes (i.e. two k-graphlets) u and v, directed from u to v, iff u has a smaller number of edges than v and $EHD(u, v) = d$.

Definition 11. *Define a* ***k-Graphlet Edge Hamming Distance 1 Sequence*** *to be a sequence of k-graphlets* $\{g_1, g_2, ..., g_s\}$ *in which* $EHD(g_i, g_{i+1}) = 1, \forall i \in \{1, .., (s-1)\}$.

Definition 12. *Define a* ***Complete k-Graphlet Edge Hamming Distance*** 1 ***Sequence*** *to be a sequence of k-graphlets* $\{g_1, g_2, ..., g_s\}$ *in which* $EHD(g_i, g_{i+1}) = 1, \forall i \in \{1, .., s\}$, *where* $s = \binom{k}{2} - (k-1) + 1$. *We show later that this is an ordered sequence of k-graphlets beginning with a spanning tree and ending with the clique on k nodes.*

Definition 13. *Define the* ***depth*** *of a node v in* H_d *to be the minimum length of the paths from v to any node with in-degree* 0 *in* H_d.

For a given value of k, consider H_1, the graph of k-graphlets in which two graphlets (nodes) are connected if they have EHD of 1. The structure of H_1 gives rise to a hierarchy among the nodes.

Lemma 2. *The nodes of* H_1 *with no incoming edges are the k-graphlets which are spanning trees.*

Proof. First, we observe that if g_1 and g_2 both have the same number of edges then $EHD(g_1, g_2) > 1$. This is because we would need to remove at least one edge and add an edge to transform g_1 into g_2. Hence, graphlets which contain the same number of edges are not neighbors of each other in H_1.

It is straightforward to see that the k-graphlets which are spanning trees have no incoming edges. Edges in H_1 exist between any pair of nodes g_1, g_2 for which $EHD(g_1, g_2) = 1$ and by

convention, we assume g_1 has fewer nodes than g_2. Since no k-graphlet contains fewer edges than $k-1$ edges, it means that each spanning tree graphlet has no incoming edge.

Finally, we show that there is no other node in H_1 with no incoming edges. Suppose graphlet g contained more than $k-1$ edges. Let ST be any spanning tree of g. By definition, ST is connected and has $k-1$ edges. This means ST is a node of H_1 which has no incoming edges. In addition, $EHD(ST, g) > 1$. Let $R = \{e_1, e2, ...e_{EHD(ST,g)}\}$ be the set of edges of g which are not in ST. If we place the edges of R one by one into ST to form g, we would form an EHD 1 sequence of graphlets: $ST, ST \cup e_1, ..., e_{EHD(ST,g)}, g$. This means that g has an incoming edge in H_1. Hence, only graphlets which are spanning trees have no incoming edges in H_1. □

Corrollary 3. *Graphlets with the same number of edges are not neighbors in H_1.*

Lemma 3. *There is only one leaf node of H_1 and it is the clique on k-nodes.*

Proof. It is easy to see that the clique on k-nodes must be a leaf node of H_1. Since edges are directed towards graphlets with more edges, and there is no graphlet with more edges than a clique, it means the clique has no children nodes and hence, must be a leaf node of H_1.

It remains to be shown that there are no other nodes of H_1 which are leaf nodes. Suppose g is a k-graphlet which is not a clique. Then, $EHD(g, clique) > 1$ because we can add edges to g to make it into a clique. Hence, g must have an outgoing edge and therefore cannot be a leaf node of H_1. □

Lemma 4. *H_1 has a depth of $\binom{k}{2} - (k-1) + 1$.*

Proof. Since we have established that only spanning tree graphlets have no incoming edges in H_1 and that only the clique is a leaf node, we must show that the path from one of the tree graphlet nodes to the clique is the longest path in the H_1. We can add $\binom{k}{2} - (k-1)$ edges iteratively to any tree graphlet g to turn it into the clique on k nodes, c. By doing

this, we form a Complete EHD 1 Sequence from g to c and this is a shortest path from g to c in H_1. There is no shorter path because we must add exactly $\binom{k}{2} - (k-1)$ edges to form the clique from g. We must therefore show that there is no longer path in H_1.

Suppose a longer path $g_1, ..., g_r$ existed in $H_1, (r > \binom{k}{2} - (k-1) + 1)$. Since $EHD(g_i, g_{i+1}) = 1, \forall i \in \{1, .., r\}$, then g_r must have r more edges than g_1. Since g_1 must have at least $k-1$ edges, it means g_r must have at least $\binom{k}{2} + 1$ edges, which is a contradiction.

Hence, the longest path in H_1 is any path from the spanning tree graphlets to the clique and any of these paths have length $\binom{k}{2} - (k-1) + 1$. Hence, H_1 must have a depth of $\binom{k}{2} - (k-1) + 1$. □

Corrollary 4. *Any path from any spanning tree graphlet node in H_1 to the clique is a Complete EHD 1 Sequence.*

Corrollary 5. *Any path in H_1 forms a EHD 1 Sequence.*

Corrollary 6. *Graphlets with the same number of edges belong to the same depth in H_1.*

Proof. Consider all Complete EHD 1 Sequences in H_1, $S = \{S_1, S_2, ...S_m\}$, where $m =$ the total number of possible complete EHD 1 Sequences in H_1. Denote $S_i[j]$ to be the j^{th} graphlet in Complete EHD 1 Sequence S_i. For all j, $S_i[j]$ is at the $(j-1)^{th}$ depth of H_1 and contains $(k-1) + j$ edges. □

6.2.2 Traversing H_1 for All Complete Sequences

We can find all the Complete k-Graphlet Edge Hamming Distance 1 Sequences in H_1 via a recursive Depth First Search with backtracking, which outputs every path in the graph from the first layer of tree-nodes to the bottom layer with the clique-node. This is shown in Algorithm 5 which uses a helper function (Algorithm 6). Table 6.1 shows the number of Complete Edge Hamming Distance 1 Sequences for k-graphlets on $2, 3, 4$ and, 5 nodes.

Algorithm 5 Traversing All Complete Sequences in H_1

procedure TRAVERSETREE($pathFound$, H_1Graph)
  $listOfAllPaths \leftarrow \phi$
  for each *root* of H_1Graph **do**
    Add TraverseHelpFunc({}, *root*) to $listOfAllPaths$
  end for
  return $listOfAllPaths$
end procedure

Algorithm 6 Helper Function for TraverseTree

procedure TRAVERSEHELPFUNC(pf, *node*)
  Add *node* to pf
  if *node* has no neighbors **then**
    return pf
  end if
  for each neighbor n of *node* **do**
    Add TraverseHelpFunc(pf, n) to pf
  end for
end procedure

Number of Nodes	Number of Complete EHD 1 Sequences
2	1
3	1
4	2
5	54

Table 6.1: Number of Complete Edge Hamming Distance 1 Sequences for $2, 3, 4$ and, 5-nodes graphlets. The number increases exponentially with k. The number for $k = 6$ is well over 5000 but we did not verify that the search ended correctly and so the number is not shown here.

6.3 Results

Metric	CON	OCD
Average Degree and Standard Deviation	19.9 +/- 14.74	19.9 +/- 15.55
Average Clustering Coefficient	0.486	0.548
Global Clustering	0.513	0.478
Diameter	8	6
Average Shortest Path	2.49	1.95
Total Graphlets	18485524	24947252

Table 6.2: Some basic properties of the CON and OCD networks are shown in this table.

6.3.1 Degree Distributions

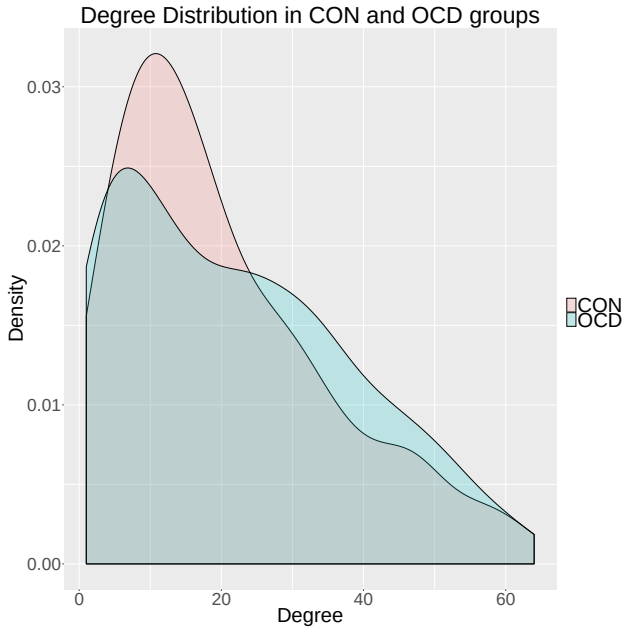


Figure 6.2: The degree distribution histograms of both OCD and CON groups. Since the two histograms are so different, it is clear that the connectivity in both groups is different. OCD has more nodes with high degrees over 40 than CON, while CON has more nodes with degree between $5 - 20$. This suggests the presence of hub nodes in OCD.

Figure 6.2 shows the degree distribution as histograms for both CON and OCD networks. OCD has more nodes with high degrees (≥ 40) than CON while CON has more nodes with degree between $5 - 20$. This suggests the presence of hub nodes in OCD. Figure 6.3 shows

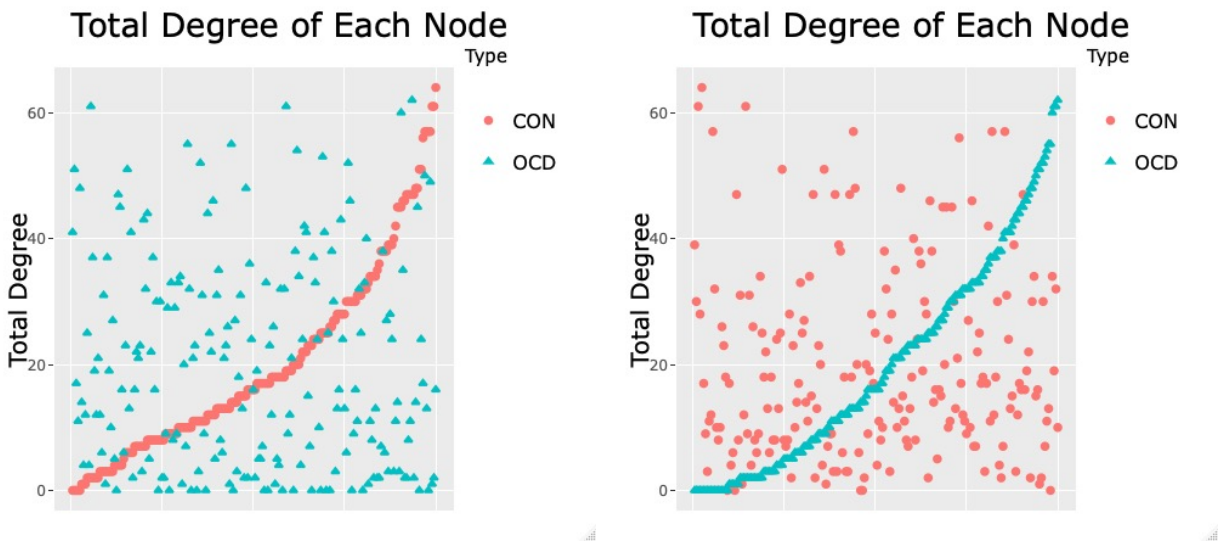


Figure 6.3: **Left:** After re-ordering the nodes of CON by their degree and comparing with the corresponding node of OCD, we observe that there is no obvious relationship between the degree of corresponding nodes in either network. **Right:** After re-ordering the nodes of OCD by their degree and comparing the corresponding node in the CON network, we make the same observation.

the node-by-node degree comparison in both networks, ordered by the degrees in the CON network and also ordered by the degrees in the OCD network. The node degrees in both networks show no similarity with each other. There is virtually no correlation between the degree distributions of both networks as their correlation coefficient is 10^{-3}. This means that different regions of the brain are co-activated very differently in OCD patients than they are in the group of CON.

6.3.2 Graphlet Distributions

The overall total number of graphlets present in the CON network is 18485524 but there is a total of 24947252 graphlets in OCD, i.e. 1.3 times the number of graphlets of CON (see Table 6.2). Figure 6.4 shows the ratio of the total number of graphlets in OCD to the total number of grahplets in CON, for each type of graphlet. We observe that the ratios are over 1 for most graphlets and hence, in the OCD network and there is a greater variety and

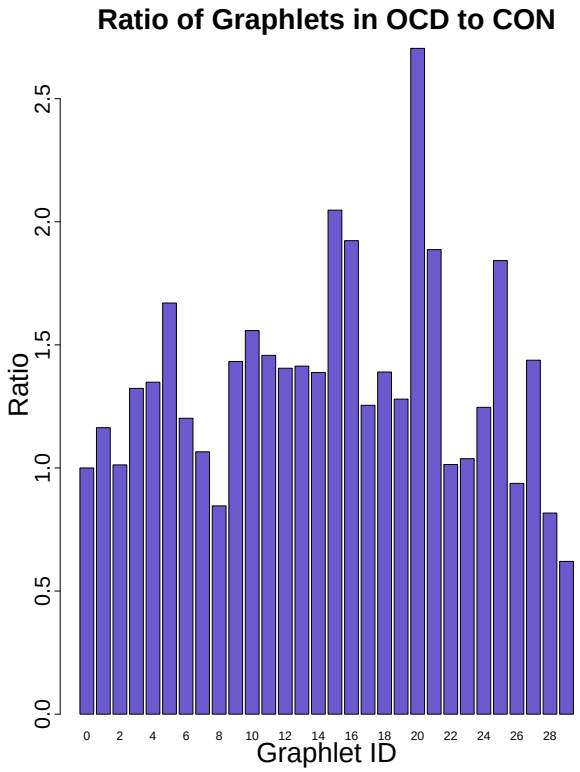


Figure 6.4: Ratio of total graphlets in OCD to total graphlets in CON, for each type of graphlet is shown here. The ratio is over 1 for all graphlets except G_8, G_{26}, G_{28} and, G_{29}. This means in general, the OCD network has a greater number of graphlets than CON.

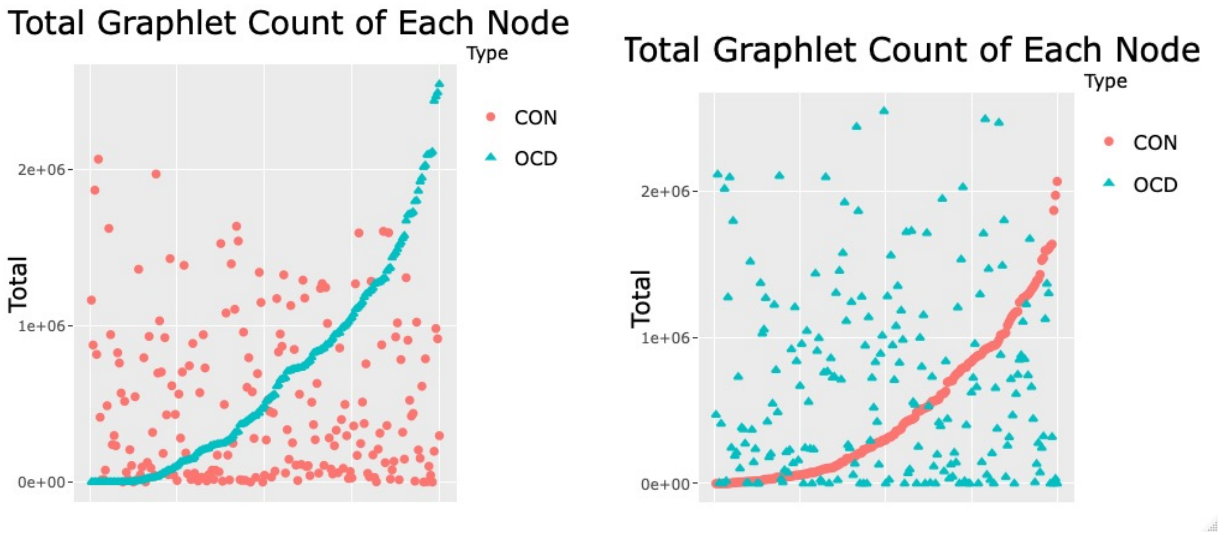


Figure 6.5: Left: The total number of graphlets (total graphlet degree) at each node in OCD. Right: The total number of graphlets (total graphlet degree) at each node in CON. In both images, the nodes are shown in order of increasing graphlet degree. The corresponding nodes from CON are plotted alongside. We see that some OCD nodes have a much higher graphlet degree than corresponding nodes in CON. This also suggests that there are particular nodes in OCD which may be hub nodes playing important roles in connecting the network.

number of most graphlets than in the CON network. This means the patterns of co-activity in the OCD brain are very different and more varied to the patterns in CON. There are only 4 graphlet types for which OCD has fewer graphlets: G_8, G_{26}, G_{28} and, G_{29}. In particular, G_8 and G_{29} are both cliques of 4 and 5 nodes respectively, and G_{26} and G_{28} are very close in topology to the 5-node clique (i.e. G_{29}). Though there are areas of high density in OCD, they are not as prevalent as in CON, despite both networks having the same overall density of 0.1. This is corroborated by the global clustering coefficient of both networks: 0.513 in CON and 0.478 in OCD (see Table 6.2). The higher global clustering coefficient implies there are many more triangles in the CON network than the OCD network relative to the number of connected node triples. This is partially evidenced by the ratio of cliques. Another noteworthy observation is that the ratio spikes at G_5, G_{15} and, G_{20}, i.e. the square, pentagon and square connected to a $5th$ node by 2 edges (Figure 6.4). This means cycles are much more common in the OCD network than in CON. Figure 6.5 also shows the node-by-node total graphlet degree ordered according to OCD and clearly shows that many nodes in OCD

contain more graphlets than the corresponding nodes in CON.

Figures 6.6 and 6.7 show the graphlet distribution in both networks for clique-like graphlets G_8, G_{26}, G_{28} and, G_{29} and cycle-like graphlets G_5, G_{15} and, G_{20}. CON has more nodes which belong to a large number of clique-like graphlets while OCD has more nodes belonging to a smaller number of cliques. Parallel to this, OCD has more nodes belonging to a large number of cycle-like graphlets while CON has more nodes belonging to a smaller number of cycle-like graphlets.

6.3.3 Graphlets on Complete Edge Hamming Distance 1 Sequences

We observe in Figure 6.4 that the ratio of 4-graplhets decreases as the density of the graphlet increases. Similarly, the ratio of 5-graphlets generally decreases as the density of graphlet increases. To quantify this observation, we computed the linear correlation coefficient along every possible Complete Edge Hamming Distance 1 Sequence of 5-graphlets and after doing a Bonferroni correct, we found the linear correlations to be statistically signficant with $p < 10^{-3}$. Figure 6.8 shows 12 randomly selected sequences from the 54 Edge Hamming Distance-1 Sequences on 5-graphlets and their linear correlation coefficients. We do not show the sequences for 4-graphlets as there are only 2 of them and they overlap on half of the graphlets.

6.3.4 Distances

Table 6.2 shows the diameter of both networks as well as their average shortest path lengths. CON has a diameter of 8 while OCD has a diameter of 6. The average shortest path length is also greater in the CON network than in the OCD network. Tables 6.3 contain the nodes in CON and OCD whose shortest paths to every other node in the network correlated with

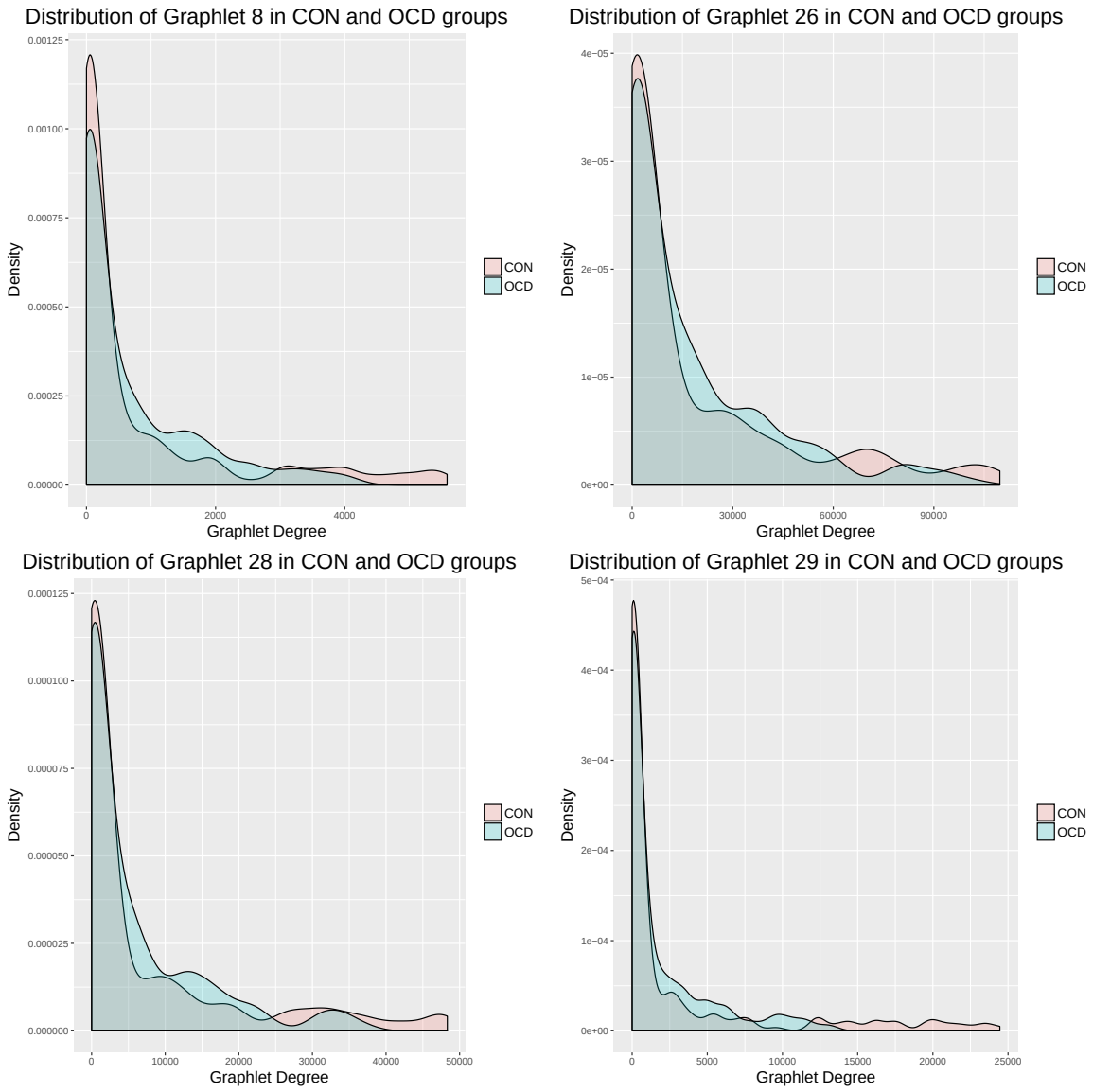


Figure 6.6: The graphlet distributions for graphlets G_8, G_{26}, G_{28} and, G_{29} are shown above. These graphlets are either cliques or very close to cliques in structure. We observe that overall, the OCD network contains more nodes with lower degrees for these graphlets when compared with the CON network, but there are more nodes in the CON network than the OCD network with higher graphlet degress for these particular graphlets.

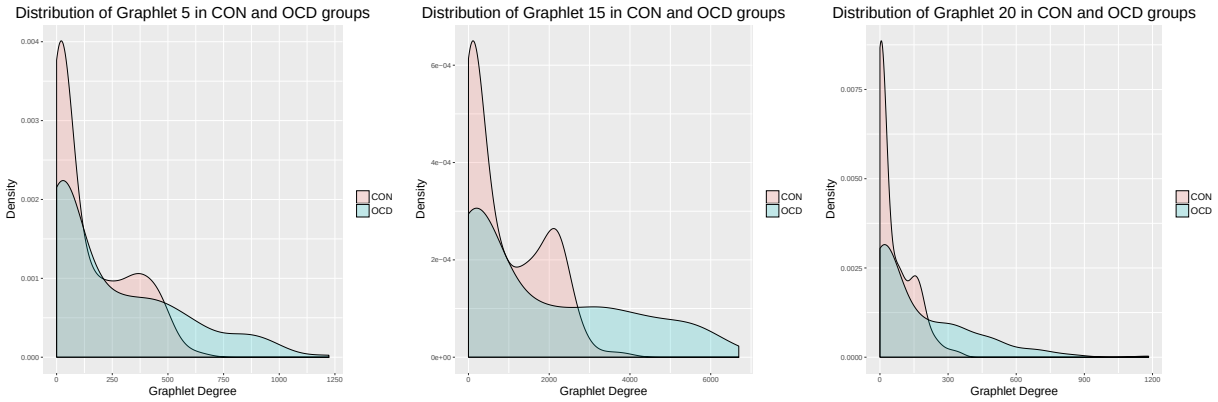


Figure 6.7: The graphlet distributions for graphlets G_5, G_{15}, and G_{20} are shown above. These graphlets are either polygons or close to polygons in their structure. We observe that overall, the OCD network contains more nodes with higher degrees for these graphlets when compared with the CON network, but there are more nodes in the CON network than the OCD network with lower graphlet degrees for these particular graphlets.

their physical distances, with a correlation coefficient of greater than 0.3. There were no negative correlations. The maximum correlation in CON was 0.344 whereas in OCD, the maximum was much higher, at 0.533.

6.3.5 Hubs

Both networks have hub nodes (i.e. nodes whose degree are much higher than the average degree in the network). However, as Table 6.4 shows, the hub nodes are different and almost of equal degree in both networks. This is also a clear sign of altered connectivity in both networks. The degrees of these hub nodes are more than 2 standard deviations away from the average degree in the networks. As their degrees are over 60, they are connected to more than a quarter of the nodes in their respective networks. The hub nodes in CON are completely different from the hub nodes of the OCD network. This means the simultaneous activation patterns are very different in the two brains.

Table 6.5 shows the top 9 nodes from the CON and OCD networks which have the highest overall graphlet degree, i.e. the nodes which belong to the highest number of graphlets.

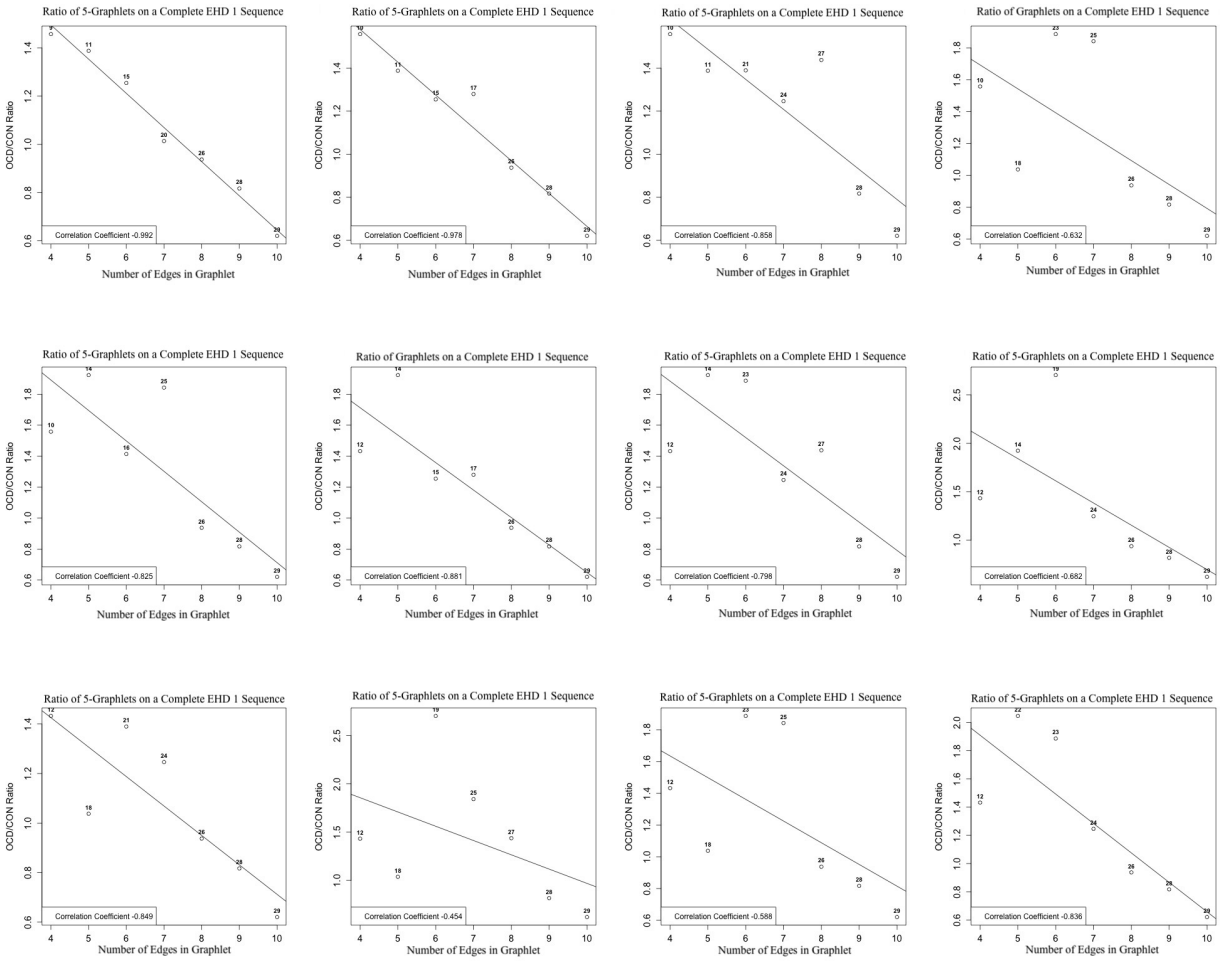


Figure 6.8: *EHD*-1 Sequences are shown above for 12 randomly chosen sequences out of all 54 sequences. Some of the correlations are high, implying a very linear change in ratio of graphlets in OCD to CON within a particular sequence. This further corroborates our finding that there are fewer dense graphlets in OCD patients than in healthy controls. The overall connectivity is lower in OCD than in CON.

Network	**Node**	**Correlation**
CON	R Posterior Temporal Fusiform Ctx Anterior Parahippocampal Gyrus Posterior ITG 131	0.303
	L Posterior Temporal Fusiform Ctx Anterior Parahippocampal Gyrus 74	0.305
	R Anterior ITG 171	0.311
	M AMCC 103	0.313
	R BS 97	0.317
	L Frontal Pole 145	0.327
	L Frontal Pole 161	0.344
OCD	R Inferior Lateral Occipital Ctx 101	0.307
	R Inferior Lateral Occipital Ctx Occipital Fusiform Gyrus 123	0.309
	R Occipital Pole Superior Lateral Occipital Ctx 174	0.313
	L Occ Fusiform Gyrus Lingual Gyrus 162	0.315
	R Lingual Gyrus Precuneous Ctx vPCC 67	0.332
	R cereb 166	0.345
	R TOFFC 168	0.356
	M Lingual Gyrus Occ Pole 172	0.358
	L cereb 61	0.373
	R cereb 183	0.378
	R BS Parahippocampal Gyrus Posterior 73	0.387
	L cereb 185	0.388
	L cereb 126	0.396
	L cereb 34	0.421
	R Lingual Gyrus Occipital Fusiform Gyrus 179	0.425
	L cereb 51	0.447
	L cereb 56	0.454
	R cereb 165	0.483
	R cereb 10	0.533

Table 6.3: This table shows the nodes in the CON and OCD networks whose shortest paths to all other nodes in the network are most correlated with their physical distances.

Network	Hub Node	Degree
CON	R Frontal Pole 50	61
	L Anterior MTG Posterior MTG 54	61
	L Temporal Pole 59	64
OCD	M precuneous 181	60
	L SFG 11	61
	M precentral 118	61
	M PMCC 187	62

Table 6.4: Hub nodes from both networks are shown above. They all have degree over 60, i.e. more than 2.7 and 2.5 standard deviations from the average degree in the CON and OCD networks respectively.

Some of the nodes from OCD are also hub nodes but none of the nodes from CON are hub nodes Table 6.4.

Network	Node	Total Graphlet Degree
CON	M Frontal Medial Ctx 114	1430741
	BS 116	1543490
	R Superior Lateral Occipital Ctx Angular Gyrus 71	1595374
	L Frontal Pole 1	1596730
	R Frontal Pole 133	1605583
	L Frontal Pole 122	1624466
	L Posterior MTG 32	1638093
	L Anterior MTG Posterior MTG 54	1869288
	R Frontal Pole 50	1972693
	L Temporal Pole 69	2067601
OCD	L Superior Lateral Occipital Ctx 2	2016714
	R Superior Lateral Occipital Ctx Angular Gyrus 71	2027880
	R Posterior STG Planum Temporale Central Opercular Ctx 152	2095305
	L Precentral MFG 88	2096372
	L Postcentral Precentral 124	2104639
	R ITG temporooccipital part MTG temporooccipital part 64	2114557
	L SFG 11	2440567
	M precentral 118	2468934
	M precuneous 181	2493897
	M PMCC 187	2547925

Table 6.5: The nodes which have the highest total graphlet degrees are listed in the table above.

6.3.6 Common Topology

There are exactly 462 edges that are the same in both networks and they occur between the 28 nodes listed in Table 6.6.

6.4 Modeling fMRI OCD and CON Brain Networks

We also attempted to find an appropriate theoretical network model for the OCD and CON networks. For motivation on modeling the brain, we direct the reader to [Bullmore and Bassett, 2011]. One study attempted to model brain networks created from diffusion tensor

imaging in adults and found SFGD and STICKY networks to be the most suitable but did not use graphlet analysis [Li et al., 2013]. We attempt to model both OCD and CON brain of children.

Similar to the approach with PPI networks, we created 50 synthetic versions of each of both the CON and OCD fMRI networks using each of the following models: ER, ERDD, SF, GEO, STICKY, SFGD and GEOGD. To generate ER, SF and GEO networks, we set the relevant parameters (such as number of nodes, density, radius, attachment index, etc) so that the resulting graphs matched the size and density of each of the brain networks. We generated 9 sets of SFGD networks for different values of p ranging from 0.1 to 0.9 in increments of 0.1. For each of these sets, we exactly matched the number of nodes of the original PPI network, and did a binary search on the corresponding q value until the synthetic graph contained within 1% of the number of edges in the real network, as in Pržulj et al. [2010a]. We generated GEOGD synthetic networks using both the expansion and probability cutoff methods described in Pržulj et al. [2010a], incrementing the probability by 0.1 from 0.1 to 0.9. Therefore, we created 50 networks from each of ER, ERDD, SF, GEO and, STICKY models, 450 SFGD networks, and 500 GEO-GD networks, for a total of $(250 + 450 + 500) = 1200$ synthetic networks for each network, i.e. a grand total of 2400 synthetic networks.

We ran ORCA to count all of the graphlets of size $k = 2, 3, 4$, and 5 from the OCD and CON networks as well as all their associated synthetic networks. We examined and compared the graphlet distributions to determine how similar the synthetic networks were to the originals.

6.5 Results

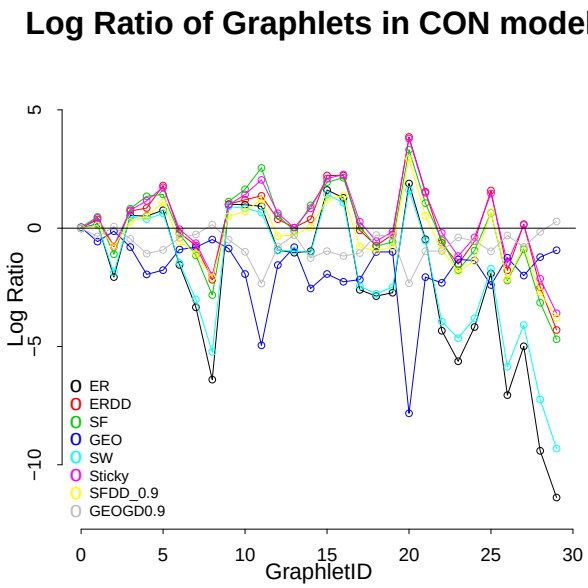
Suppose G is a network with graphlet count given by $f_G : \mathbb{N} \mapsto \mathbb{Z}$, $f_G(g)$ represents the number of graphlets of type g present in G. Figure 6.9 shows the average ratio $log(\frac{f_G(g)}{f_{REAL}(g)})$

over 50 synthetic networks, for the 30 graphlets, of sizes 2, 3, 4 and 5, for each model. In order to de-clutter the plot, we choose include only the best-fitting SFGD and GEOGD model in Figure 6.9. Table 6.7 quantifies these observations, corroborating the visual observation. It is the 'area under the curve' (AUC) for all models, i.e. a measure of how much the graphlet distribution deviates from the true distribution of the real network.

6.6 Conclusion and Discussion

Our findings from degree distribution and graphlet analysis on the OCD and CON networks show that the brain activation patterns in both networks are entirely different. Moreover, the nodes that are most co-activvated in both networks are completely different from each other. We find that the circuit connections in the OCD network are much more prevalent while cliques are more prevalent in the CON network. The utility of the hub nodes in the function of the brain could explain behavioural differences in OCD patients from healthy controls, because hub nodes may be most frequently (or most easily) activated, but this is left as future work.

In addition, we have modeled the OCD and CON networks and find that the CON network most closely resembles a GEOGD network compared to the other models. However, the best matching model was less clear for the OCD network. STICKY matched most, followed very closely by ERDD and then SFGD. This could indicate that the OCD network has mixed features of all of these network models or that we have yet to find the right model for OCD rsfMRI connectivity.



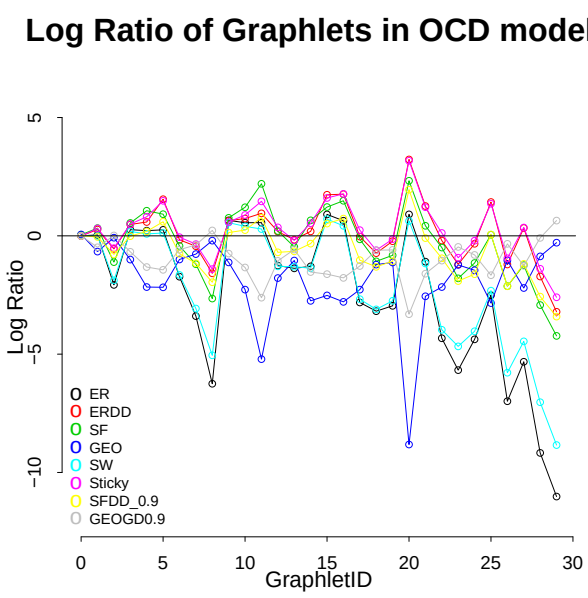


Figure 6.9: The average log ratio over 50 synthetic networks, for each graphlet g is shown above, i.e. $mean(log(\frac{f_G(g)}{f_{REAL}(g)}))$. An eyeball inspection suggests that, in order of best to worst, Sticky, ERDD and $SFDD_{0.9}$ models of the OCD network 'hug' the x-axis most closely compared to other models, and the GEOGD model of the CON network most closely 'hugs' the x-axis. This observation is quantified in Table 6.7.

Nodes Supporting Common Edges in Both Networks
L Superior Lateral Occipital Ctx 2
R SFG 120
R Inferior Lateral Occipital Ctx Occipital Fusiform Gyrus 123
L Anterior Parahippocampal Gyrus Temporal Fusiform Ctx 35
L Inferior Lateral Occipital Ctx Occipital Fusiform Gyrus 147
R Frontal Pole FOC 150
R Posterior STG Planum Temporale Central Opercular Ctx 152
R Posterior STG Posterior MTG 17
L Temporal Pole Anterior ITG 163
R Precentral Central Oper Ctx IFG pars oper 164
R cereb 165
R cereb 166
R Occipital Pole Superior Lateral Occipital Ctx 174
R IFG pars oper 178
R Lingual Gyrus Occipital Fusiform Gyrus 179
R cereb 19
L Thalamus Caudate 182
R cereb 183
L SFG 184
L cereb 22
L IFG pars oper 24
L Occipital Pole Superior Lateral Occipital Ctx 40
R MTG temporooccipital part 5
L cereb 51
R Postcentral Precentral 58
L Occipital Pole Occipital Fusiform Gyrus 79
R cereb 87
L Planum Polare Temporal Pole Insular Ctx Central Opercular Ctx 91

Table 6.6: The edges which are common in both networks occur between only these 28 nodes.

Network Model	CON	OCD
ER	83.57	81.79
ERDD	35.99	**27.09**
SF	41.07	35.27
GEO	51.19	55.68
SW	71.52	69.82
Sticky	34.64	**26.09**
$SFDD_{0.1}$	60.90	57.08
$SFDD_{0.2}$	49.36	45.38
$SFDD_{0.3}$	42.37	38.44
$SFDD_{0.4}$	39.74	34.92
$SFDD_{0.5}$	37.89	31.03
$SFDD_{0.6}$	34.82	30.44
$SFDD_{0.7}$	33.98	29.27
$SFDD_{0.8}$	33.05	28.91
$SFDD_{0.9}$	32.39	**28.86**
GEOGD0.1	29.14	38.77
GEOGD0.2	28.64	36.00
GEOGD0.3	26.25	33.52
GEOGD0.4	22.51	33.20
GEOGD0.5	21.37	34.71
GEOGD0.6	24.83	30.89
GEOGD0.7	22.28	30.14
GEOGD0.8	21.80	33.28
GEOGD0.9	**21.39**	30.10

Table 6.7: The area under the curves are shown here for each model, including all the SFDD and GEOGD models not shown in Figure 6.9. GEOGD0.9 is by far the closest fit for the CON network from amongst these 7 network types. the OCD network is fit best by STICKY, ERDD and $SFDD_{0.9}$, in order from best to worst.

Chapter 7

Future Work

7.1 Modeling Networks

7.1.1 PPI Networks

As mentioned in chapter 2, though there have been significant updates to the BioGRID PPI networks, they are still quite sparse and incomplete. This problem of modeling the structure of PPI networks should be revisited again when the data are much more correct and complete. It would be also be useful to model other larger PPI networks from other sources, to observe if the same models are consistent across different datasets acquired from different bio-technologies.

7.1.2 OCD Brain Networks

In chapter 6, we modeled one brain OCD network but could not definitively match the structure to any one model. Other kinds of models should be considered or created to

specifically match brain network patterns in future studies of this nature. In addition, the same exploration done in chapter 6 should be performed on other OCD and control networks of other studies to ensure consistency in results. The clinical relevance of all these findings should also be explored.

7.2 Sampling Graphlets

As noted in chapter 3, in order to ensure correctness and quality of results reproduced by graphlet sampling, it is necessary to have an accurate node-by-node estimate of the number of each graphlet and orbit in a network. In addition, we seek to correct the bias of Node Based Expansion by deriving the theoretical expected distribution of graphlets that is obtained by sampling via NBE.

Bibliography

N. K. Ahmed, N. Duffield, J. Neville, and R. Kompella. Graph sample and hold: A framework for big-graph analytics. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1446–1455. ACM, 2014.

N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield. Efficient graphlet counting for large networks. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 1–10. IEEE, 2015.

W. Ali, T. Rito, G. Reinert, F. Sun, and C. M. Deane. Alignment-free protein interaction network comparison. *Bioinformatics*, 30(17):i430–i437, 2014.

A. Anticevic, S. Hu, S. Zhang, A. Savic, E. Billingslea, S. Wasylink, G. Repovs, M. W. Cole, S. Bednarski, J. H. Krystal, et al. Global resting-state functional magnetic resonance imaging analysis identifies frontal cortex, striatal, and cerebellar dysconnectivity in obsessive-compulsive disorder. *Biological psychiatry*, 75(8):595–605, 2014.

A. APA. Diagnostic and statistical manual of mental disorders-dsm-iv-tr (vol. text revision), 1994.

C. C. Armstrong, T. D. Moody, J. D. Feusner, J. T. McCracken, S. Chang, J. G. Levitt, J. C. Piacentini, and J. O'Neill. Graph-theoretical analysis of resting-state fmri in pediatric obsessive–compulsive disorder. *Journal of affective disorders*, 193:175–184, 2016.

A. Barabási, Z. Dezso, E. Ravasz, Z.-H. Yook, and Z. N. Oltvai. Scale-free and hierarchical structures in complex networks. In *Modeling of Ccomplex Systems: Seventh Granada Lectures. AIP Conference Proceedings*, volume 661, pages 1–16, 2003.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439): 509–512, 1999.

M. G. Bell and Y. Iida. *Transportation network analysis*. 1997.

M. A. Bhuiyan, M. Rahman, M. Rahman, and M. Al Hasan. Guise: Uniform sampling of graphlets for large graph analysis. In *2012 IEEE 12th International Conference on Data Mining*, pages 91–100. IEEE, 2012.

G. Bianconi, P. Pin, and M. Marsili. Assessing the relevance of node features for network structure. *Proceedings of the National Academy of Sciences*, 106(28):11433–11438, 2009.

S. Bornholdt and H. G. Schuster. *Handbook of graphs and networks: from the genome to the internet*. John Wiley & Sons, 2006.

B.-J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bahler, V. Wood, K. Dolinski, and M. Tyers. The BioGRID Interaction Database: 2008 update. *Nucl. Acids Res.*, 36(suppl1):D637–640, 2008. doi: 10.1093/nar/gkm1001.

D. L. Brown. Modeling, simulation and analysis of complex networked systems: A program plan for doe office of advanced scientific computing research. Technical report, Lawrence Livermore National Laboratory (LLNL), Livermore, CA, 2009.

J. A. Brown and J. D. Van Horn. Connected brains and minds—the umcd repository for brain connectivity matrices. *Neuroimage*, 124:1238–1241, 2016.

J. A. Brown, J. D. Rudie, A. Bandrowski, J. D. Van Horn, and S. Y. Bookheimer. The ucla multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis. *Frontiers in neuroinformatics*, 6:28, 2012.

E. T. Bullmore and D. S. Bassett. Brain graphs: graphical models of the human brain connectome. *Annual review of clinical psychology*, 7:113–140, 2011.

T. M. Cabrera-Vera, J. Vanhauwe, T. O. Thomas, M. Medkova, A. Preininger, M. R. Mazzoni, and H. E. Hamm. Insights into g protein structure, function, and regulation. *Endocrine reviews*, 24(6):765–781, 2003.

D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM computing surveys (CSUR)*, 38(1):2, 2006.

D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-mat: A recursive model for graph mining. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 442–446. SIAM, 2004.

A. Chatr-Aryamontri, B.-J. Breitkreutz, S. Heinicke, L. Boucher, A. Winter, C. Stark, J. Nixon, L. Ramage, N. Kolas, L. O'Donnell, et al. The biogrid interaction database: 2013 update. *Nucleic acids research*, 41(D1):D816–D823, 2012.

A. Chatr-Aryamontri, R. Oughtred, L. Boucher, J. Rust, C. Chang, N. K. Kolas, L. O'Donnell, S. Oster, C. Theesfeld, A. Sellam, et al. The biogrid interaction database: 2017 update. *Nucleic acids research*, 45(D1):D369–D379, 2017.

X. Chen, Y. Li, P. Wang, and J. Lui. A general framework for estimating graphlet statistics via random walk. *Proceedings of the VLDB Endowment*, 10(3):253–264, 2016.

A. Clauset, E. Tucker, and M. Sainz. The colorado index of complex networks, 2016.

S. R. Collins, P. Kemmeren, X.-C. Zhao, J. F. Greenblatt, F. Spencer, F. C. P. Holstege, J. S. Weissman, and N. J. Krogan. Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. *Molecular and Cellular Proteomics*, 6(3):439–450, 2007. doi:

10.1074/mcp.M600381-MCP200. URL http://www.mcponline.org/content/6/3/439.abstract.

S. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd Ann. ACM Symp. on Theory of Computing: 1971; New York*, pages 151–158. Assosiation for Computing Machinery, 1971.

D. Davis, Ö. N. Yaveroğlu, N. Malod-Dognin, A. Stojmirovic, and N. Pržulj. Topology-function conservation in protein–protein interaction networks. *Bioinformatics*, 31(10): 1632–1639, 2015. doi: 10.1093/bioinformatics/btv026.

T. A. Davis and Y. Hu. The university of florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1, 2011.

M. Dehmer, F. Emmert-Streib, and Y. Shi. Interrelations of graph distance measures based on topological indices. *PloS one*, 9(4):e94985, 2014.

D. M. Dunlavy, B. Hendrickson, and T. G. Kolda. Mathematical challenges in cybersecurity. *Sandia Report, February*, 2009.

F. Emmert-Streib, M. Dehmer, and Y. Shi. Fifty years of graph matching, network alignment and network comparison. *Information Sciences*, 346:180–197, 2016.

P. Erdös and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

F. E. Faisal, L. Meng, J. Crawford, and T. Milenković. The post-genomic era of biological network alignment. *EURASIP Journal on Bioinformatics and Systems Biology*, 2015(1): 1, 2015.

K. D. Fitzgerald, E. R. Stern, M. Angstadt, K. C. Nicholson-Muth, M. R. Maynor, R. C. Welsh, G. L. Hanna, and S. F. Taylor. Altered function and connectivity of the medial frontal cortex in pediatric obsessive-compulsive disorder. *Biological psychiatry*, 68(11): 1039–1047, 2010.

T. Furusawa and H. Konishi. Free trade networks. *Journal of International Economics*, 72 (2):310–335, 2007.

D. Garlaschelli and M. I. Loffredo. Structure and evolution of the world trade network. *Physica A: Statistical Mechanics and its Applications*, 355(1):138–144, 2005.

A. Gutfraind, I. Safro, and L. A. Meyers. Multiscale network generation. In *Information Fusion (Fusion), 2015 18th International Conference on*, pages 158–165. IEEE, 2015.

B. J. Harrison, C. Soriano-Mas, J. Pujol, H. Ortiz, M. López-Solà, R. Hernández-Ribas, J. Deus, P. Alonso, M. Yücel, C. Pantelis, et al. Altered corticostriatal functional connectivity in obsessive-compulsive disorder. *Archives of general psychiatry*, 66(11):1189–1200, 2009.

A. Hasan, P.-C. Chung, and W. Hayes. Graphettes: Constant-time determination of graphlet and orbit identity including (possibly disconnected) graphlets up to size 8. *PloS one*, 12 (8):e0181570, 2017.

W. Hayes and S. Maharaj. Blant: Sampling graphlets in a flash.

W. Hayes, K. Sun, and N. Pržulj. Graphlet-based measures are suitable for biological network comparison. *Bioinformatics*, 29(4):483–491, 2013.

I. Heyman, E. Fombonne, H. Simmons, T. Ford, H. Meltzer, and R. Goodman. Prevalence of obsessive-compulsive disorder in the british nationwide survey of child mental health. *International Review of Psychiatry*, 15(1-2):178–184, 2003.

D. Higham, M. Rašajski, and N. Pržulj. Fitting a geometric graph to a protein-protein interaction network. *Bioinformatics*, 24(8):1093–1099, 2008.

T. Hočevar and J. Demšar. Combinatorial algorithm for counting small induced graphs and orbits. *PloS one*, 12(2):e0171428, 2017.

F. Hormozdiari, P. Berenbrink, N. Pržulj, and C. Sahinalp. Not all scale free networks are born equal: the role of the seed graph in ppi netwok emulation. in press.

J.-M. Hou, M. Zhao, W. Zhang, L.-H. Song, W.-J. Wu, J. Wang, D.-Q. Zhou, B. Xie, M. He, J.-W. Guo, et al. Resting-state functional connectivity abnormalities in patients with obsessive–compulsive disorder and their healthy first-degree relatives. *Journal of psychiatry & neuroscience: JPN*, 39(5):304, 2014.

T. Hočevar and J. Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, Feb. 2014. ISSN 1460-2059. doi: 10.1093/bioinformatics/btt717. URL `http://dx.doi.org/10.1093/bioinformatics/btt717`.

R. Ibragimov, M. Malek, J. Guo, and J. Baumbach. Gedevo: an evolutionary graph edit distance algorithm for biological network alignment. In *OASIcs-OpenAccess Series in Informatics*, volume 34. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

R. Ibragimov, M. Malek, J. Baumbach, and J. Guo. Multiple graph edit distance: simultaneous topological alignment of multiple protein-protein interaction networks with an evolutionary algorithm. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 277–284. ACM, 2014.

V. Janjić and N. Pržulj. The topology of the growing human interactome data. *Journal of integrative bioinformatics*, 11(2):27–42, 2014.

V. Janjić, R. Sharan, and N. Pržulj. Modelling the yeast interactome. *Scientific reports*, 4, 2014.

M. Jha, C. Seshadhri, and A. Pinar. Path sampling: A fast and provable method for estimating 4-vertex subgraph counts. In *Proceedings of the 24th International Conference on World Wide Web*, pages 495–505. International World Wide Web Conferences Steering Committee, 2015.

G. Karlebach and R. Shamir. Modelling and analysis of gene regulatory networks. *Nature reviews. Molecular cell biology*, 9(10):770, 2008.

N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.

B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *PNAS*, 100:11394–11399, 2003.

R. Khanin and E. Wit. How scale-free are biological networks. *Journal of computational biology*, 13(3):810–818, 2006.

J. Knopp, S. Knowles, P. Bee, K. Lovell, and P. Bower. A systematic review of predictors and moderators of response to psychological therapies in ocd: Do we have enough empirical evidence to target treatment? *Clinical psychology review*, 33(8):1067–1081, 2013.

T. G. Kolda, A. Pinar, T. Plantenga, and C. Seshadhri. A scalable generative graph model with community structure. *SIAM Journal on Scientific Computing*, 36(5):C424–C452, 2014.

E. Kross, P. Verduyn, E. Demiralp, J. Park, D. S. Lee, N. Lin, H. Shablack, J. Jonides, and O. Ybarra. Facebook use predicts declines in subjective well-being in young adults. *PloS one*, 8(8):e69841, 2013.

O. Kuchaiev and N. Pržulj. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396, 2011a. URL `http://www.ncbi.nlm.nih.gov/pubmed/21414992`.

O. Kuchaiev and N. Pržulj. Integrative network alignment reveals large regions of global network similarity in yeast and human. *BIOINFORMATICS*, 27:1390–1396, 2011b. doi: bioinformatics/btr127. URL `http://dx.doi.org/10.1093/bioinformatics/btr127`.

O. Kuchaiev, T. Milenković, V. Memišević, W. Hayes, and N. Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of The Royal Society Interface*, 7(50):1341–1354, 2010. ISSN 1742-5689. doi: 10.1098/rsif.2010.0063.

C. W. Lack, E. A. Storch, M. L. Keeley, G. R. Geffken, E. D. Ricketts, T. K. Murphy, and W. K. Goodman. Quality of life in children and adolescents with obsessive-compulsive disorder: base rates, parent–child agreement, and clinical correlates. *Social Psychiatry and Psychiatric Epidemiology*, 44(11):935–942, 2009.

J. Leskovec and R. Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1, 2016.

J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11 (Feb):985–1042, 2010.

X. Li, X. Hu, C. Jin, J. Han, T. Liu, L. Guo, W. Hao, and L. Li. A comparative study of theoretical graph models for characterizing structural networks of human brain. *International journal of biomedical imaging*, 2013, 2013.

C.-S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–258, 2009.

X. Lu and S. Bressan. Sampling connected induced subgraphs uniformly at random. In *International Conference on Scientific and Statistical Database Management*, pages 195–212. Springer, 2012.

K. Luck, G. M. Sheynkman, I. Zhang, and M. Vidal. Proteome-scale human interactomics. *Trends in Biochemical Sciences*, 2017.

F. Luo, Y. Yang, C.-F. Chen, R. Chang, J. Zhou, and R. H. Scheuermann. Modular organization of protein interaction networks. *Bioinformatics*, 23(2):207–214, 2006.

T. V. Maia, R. E. Cooney, and B. S. Peterson. The neural bases of obsessive–compulsive disorder in children and adults. *Development and psychopathology*, 20(4):1251–1283, 2008.

N. Malod-Dognin and N. Pržulj. L-graal: Lagrangian graphlet-based network aligner. *Bioinformatics*, 31(13):2182–2189, 2015.

N. Malod-Dognin, K. Ban, and N. Pržulj. Unified alignment of protein-protein interaction networks. *Scientific Reports*, 7(1):953, 2017.

N. Mamano and W. B. Hayes. Sana: simulated annealing far outperforms many other search algorithms for biological network alignment. *Bioinformatics*, page btx090, 2017.

D. Marcus and Y. Shavitt. Rage–a rapid graphlet enumerator for large networks. *Computer Networks*, 56(2):810–819, 2012.

A. Medina, I. Matta, and J. Byers. On the origin of power laws in internet topologies. *ACM SIGCOMM computer communication review*, 30(2):18–28, 2000.

I. Melckenbeeck, P. Audenaert, T. Michoel, D. Colle, and M. Pickavet. An algorithm to automatically generate the combinatorial orbit counting equations. *PLoS ONE*, 11(1), 2016. doi: http://dx.doi.org/10.1371/journal.pone.0147078.

I. Melckenbeeck, P. Audenaert, D. Colle, and M. Pickavet. Efficiently counting all orbits of graphlets of any order in a graph using autogenerated equations. *preprint*, 2017a.

I. Melckenbeeck, P. Audenaert, D. Colle, and M. Pickavet. Efficiently counting all orbits of graphlets of any order in a graph using autogenerated equations. *Bioinformatics*, 1:9, 2017b.

L. Menzies, S. R. Chamberlain, A. R. Laird, S. M. Thelen, B. J. Sahakian, and E. T. Bullmore. Integrating evidence from neuroimaging and neuropsychological studies of obsessive-compulsive disorder: the orbitofronto-striatal model revisited. *Neuroscience & Biobehavioral Reviews*, 32(3):525–549, 2008.

D. Meunier, K. D. Ersche, K. J. Craig, A. Fornito, E. Merlo-Pich, N. A. Fineberg, S. S. Shabbir, T. W. Robbins, and E. T. Bullmore. Brain functional connectivity in stimulant drug dependence and obsessive–compulsive disorder. *Neuroimage*, 59(2):1461–1468, 2012.

C. G. M. Mihail and E. Zegura. The markov chain simulation method for generating connected power law random graphs. In *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments*, volume 111, page 16. SIAM, Philadelphia, 2003.

M. R. Milad and S. L. Rauch. Obsessive-compulsive disorder: beyond segregated cortico-striatal pathways. *Trends in cognitive sciences*, 16(1):43–51, 2012.

M. Milano, P. H. Guzzi, O. Tymofieva, D. Xu, C. Hess, P. Veltri, and M. Cannataro. An extensive assessment of network alignment algorithms for comparison of brain connectomes. *BMC bioinformatics*, 18(6):235, 2017.

T. Milenković and N. Pržulj. Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*, 6:257–273, 2008.

T. Milenković, J. Lai, and N. Pržulj. Graphcrunch: a tool for large network analyses. *BMC Bioinformatics*, 9(70), 2008.

T. Milenković, W. L. Ng, W. Hayes, and N. Pržulj. Optimal network alignment with graphlet degree vectors. *Cancer Informatics*, 9: 121–137, 06 2010. doi: 10.4137/CIN.S4744. URL `www.la-press.com/optimal-network-alignment-with-graphlet-degree-vectors-article-a2141`.

T. Milenković, H. Zhao, and F. E. Faisal. Global network alignment in the context of aging. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, BCB'13, pages 23:23–23:32, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2434-2. doi: 10.1145/2506583.2508968. URL `http://doi.acm.org/10.1145/2506583.2508968`.

R. Milo, S. S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298:824–827, 2002.

R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303:1538–1542, 2004.

M. Newman. Networks: an introduction. 2010. *United Slates: Oxford University Press Inc., New York*, pages 1–2, 2010.

M. E. Newman. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409, 2001.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2): 167–256, 2003.

B. Neyshabur, A. Khadem, S. Hashemifar, and S. S. Arab. Netal: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, 29 (13):1654–1662, 2013.

K. B. Nooner, S. J. Colcombe, R. H. Tobe, M. Mennes, M. M. Benedict, A. L. Moreno, L. J. Panek, S. Brown, S. T. Zavitz, Q. Li, et al. The nki-rockland sample: a model for accelerating the pace of discovery science in psychiatry. *Frontiers in neuroscience*, 6, 2012.

M. Penrose. *Random Geometric Graphs*. Oxford Studies in Probability, 2003.

J. Piacentini, T. S. Peris, R. L. Bergman, S. Chang, and M. Jaffer. Brief report: Functional impairment in childhood ocd: Development and psychometrics properties of the child obsessive-compulsive impact scale-revised (cois-r). *Journal of Clinical Child and Adolescent Psychology*, 36(4):645–653, 2007.

S. Pinkert, J. Schultz, and J. Reichardt. Protein interaction networks-more than mere modules. *PLoS computational biology*, 6(1):e1000659, 2010.

N. Pržulj. *Analyzing Large Biological Networks: Protein-Protein Interactions Example*. PhD thesis, University of Toronto, Canada, 2005.

N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 20:e177–e183, 2007.

N. Pržulj and W. B. Hayes. Biological network comparison using graphlet degree distributions. In *Proceedings of the 3rd International Symposium on Networks in Bioinformatics (ISNB'06), Amsterdam, the Netherlands*, 2006. Acceptance rate 20%.

N. Pržulj and D. Higham. Modelling protein-protein interaction networks via a stickiness index. *Journal of the Royal Society Interface*, 3(10):711–716, 2006.

N. Pržulj and T. Milenković. Computational Methods for Analyzing and Modeling Biological Networks. In J. Chen and S. Lonardi, editors, *Biological Data Mining*, volume To appear. CRC Press, 2009.

N. Pržulj, D. G. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004. doi: 10.1093/bioinformatics/bth436. URL `http://bioinformatics.oxfordjournals.org/content/20/18/3508.abstract`.

N. Pržulj, O. Kuchaiev, A. Stevanović, and W. Hayes. Geometric evolutionary dynamics of protein interaction networks. *Pacific Symposium on Biocomputing*, 2010a.

N. Pržulj, O. Kuchaiev, A. Stevanović, and W. Hayes. Geometric evolutionary dynamics of protein interaction networks. *Proceedings of the 2010 Pacific Symposium on Biocomputing (PSB), Big Island, Hawaii, January 4-8, 2010*, 2010b.

N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

N. Przulj. Erratum to Biological network comparison using graphlet degree distribution. *Bioinformatics*, 26(6):853–854, Mar. 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq091. URL `http://www.bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/btq091`.

N. Pržulj, D. G. Corneil, and I. Jurisica. Efficient estimation of graphlet frequency distributions in protein–protein interaction networks. *Bioinformatics*, 22(8):974–980, 2006.

M. Rahman, M. A. Bhuiyan, and M. Al Hasan. Graft: An efficient graphlet counting method for large graph analysis. *IEEE Transactions on Knowledge and Data Engineering*, 26(10): 2466–2478, 2014.

J. L. Rapoport, G. Inoff-Germain, M. M. Weissman, S. Greenwald, W. E. Narrow, P. S. Jensen, B. B. Lahey, and G. Canino. Childhood obsessive–compulsive disorder in the nimh meca study: Parent versus child identification of cases. *Journal of Anxiety Disorders*, 14 (6):535–548, 2000.

R. E. Rice, C. L. Borgman, and B. Reeves. Citation networks of communication journals, 1977–1985 cliques and positions, citations made and citations received. *Human communication research*, 15(2):256–283, 1988.

R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015a. URL `http://networkrepository.com`.

R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015b. URL `http://networkrepository.com`.

M. Salathé, R. M. May, and S. Bonhoeffer. The evolution of network topology by selective removal. *Roy.Soc.Interface*, 2:533–536, 2005.

V. Saraph and T. Milenković. Magna: maximizing accuracy in global network alignment. *Bioinformatics*, 30(20):2931–2940, 2014.

R. G. Schlösser, G. Wagner, C. Schachtzabel, G. Peikert, K. Koch, J. R. Reichenbach, and H. Sauer. Fronto-cingulate effective connectivity in obsessive compulsive disorder: A study with fmri and dynamic causal modeling. *Human Brain Mapping*, 31(12):1834–1850, 2010.

C. Seshadhri, A. Pinar, and T. G. Kolda. Triadic measures on graphs: The power of wedge sampling. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 10–18. SIAM, 2013.

N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*, pages 488–495, 2009.

D.-J. Shin, W. H. Jung, Y. He, J. Wang, G. Shim, M. S. Byun, J. H. Jang, S. N. Kim, T. Y. Lee, H. Y. Park, et al. The effects of pharmacological treatment on functional brain connectome in obsessive-compulsive disorder. *Biological psychiatry*, 75(8):606–614, 2014.

N. Sloane. Online encyclopedia of integer sequences (oeis), a. URL `http://oeis.org/A000088`.

N. Sloane. Online encyclopedia of integer sequences (oeis), b. URL `http://oeis.org/A000666`.

O. Sporns. *Networks of the Brain*. MIT press, USA, 2010.

O. Sporns. From simple graphs to the connectome: networks in neuroimaging. *Neuroimage*, 62(2):881–886, 2012.

O. Sporns and R. Kötter. Motifs in brain networks. *PLoS biology*, 2(11):e369, 2004.

C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: A general repository for interaction datasets. *Nucleic Acids Research*, 34:D535–D539, 2006.

E. R. Stern, K. D. Fitzgerald, R. C. Welsh, J. L. Abelson, and S. F. Taylor. Resting-state functional connectivity between fronto-parietal and default mode networks in obsessive-compulsive disorder. *PloS one*, 7(5):e36356, 2012.

T. Thorne and M. P. Stumpf. Graph spectral analysis of protein interaction network evolution. *Journal of The Royal Society Interface*, page rsif20120220, 2012.

T. Tokar, C. Pastrello, A. E. Rossos, M. Abovsky, A.-C. Hauschild, M. Tsay, R. Lu, and I. Jurisica. mirdip 4.1—integrative database of human microrna target predictions. *Nucleic acids research*, 46(D1):D360–D370, 2017.

M. M. Vaghi, P. E. Vértes, M. G. Kitzbichler, A. M. Apergis-Schoute, F. E. van der Flier, N. A. Fineberg, A. Sule, R. Zaman, V. Voon, P. Kundu, et al. Specific frontostriatal circuits for impaired cognitive flexibility and goal-directed planning in obsessive-compulsive disorder: evidence from resting-state functional connectivity. *Biological psychiatry*, 81(8): 708–717, 2017.

A. Vázquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *Complexus*, 1(1):38–44, 2003.

M. Vidal. How much of the human protein interactome remains to be mapped?, 2016.

V. Vijayan, V. Saraph, and T. Milenković. Magna++: Maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics*, page btv161, 2015.

P. Wang, J. Lui, B. Ribeiro, D. Towsley, J. Zhao, and X. Guan. Efficiently estimating motif statistics of large networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(2):8, 2014.

P. Wang, J. C. Lui, D. Towsley, and J. Zhao. Minfer: A method of inferring motif statistics from sampled edges. In *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*, pages 1050–1061. IEEE, 2016.

Z. Wang and J. Zhang. In search of the biological significance of modular structures in protein networks. *PLoS computational biology*, 3(6):e107, 2007.

D. Watts and S. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393 (6684), 1998a.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393: 440–442, 1998b.

R. C. Wilson and P. Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41(9):2833–2841, 2008.

N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. K. A. Stojmirovic, and N. Pržulj. Revealing the hidden language of complex networks. *Scientific reports*, 4:4547, 2014.

Ö. N. Yaveroğlu, T. Milenković, and N. Pržulj. Proper evaluation of alignment-free network comparison methods. *Bioinformatics*, 31(16):2697–2704, 2015.

T. Zhang, J. Wang, Y. Yang, Q. Wu, B. Li, L. Chen, Q. Yue, H. Tang, C. Yan, S. Lui, et al. Abnormal small-world architecture of top–down control networks in obsessive–compulsive disorder. *Journal of psychiatry & neuroscience: JPN*, 36(1):23, 2011.

Y. Zhang and J. Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

J. Zywica and J. Danowski. The faces of facebookers: Investigating social enhancement and social compensation hypotheses; predicting facebook and offline popularity from sociability and self-esteem, and mapping the meanings of popularity with semantic networks. *Journal of Computer-Mediated Communication*, 14(1):1–34, 2008.