EFFICIENT ENUMERATION OF SMALL GRAPHLETS AND ORBITS

by

Apratim Das

THESIS SUBMITTED IN PARTIAL FULLFILMENT OF
THE REQUIREMENTS OF THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER SCIENCE

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

February 2020

Dedicated

to

My Brother and Parents

Acknowledgement

I would like to express my deep sense of gratitude to my supervisor Dr. Alex Aravind for allowing me to carry out my work under his invaluable guidance. His interest in scientific problems and moral inspiration have made the current work possible. I am extremely indebted to the Chairperson, Department of Computer Science, University of Northern British Columbia (UNBC), Prince George, for providing all the necessary facilities to complete this work. Also, a special thanks is extended to Dr. Mark Dale for his expertise in this area and guiding me through my reseach and to Dr. Stephen Rader for providing some insights during the study. Finally, I would like to thank Dr. Sam Walters, Dr. Fan Jiang and Dr. Mark Dale for offering their valuable assistance as my committee members.

I am also indebted to my labmates Gurpreet Singh, Conan Veitch, Brae Stoltz, Davis Goulet, Daniel O'Reilly and Michael Lane for their support, discussion, encouragement, and pleasant company. Special thanks to the teaching and nonteaching staff members, UNBC, for their help in all possible ways.

It is a pleasure to acknowledge the good company of fellow researchers and my friends Bryan, Alam, Richard, Yuquan, Evana, Farzana, Abdul and many others.

Finally, I want to express my deepest gratitude to my family for their love, support and encouragement throughout the course of this work. A special thanks is due to my father for inculcating academic rigour in me and inspiring me to shape this thesis.

Prince George, BC
December 6, 2019 *(Apratim Das)*

List of Publications

Apratim Das, Alex Aravind and Mark Dale (2019). *Algorithm and Application for Signed Graphlets.* Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), August 27-30, 2019, Vancouver, Canada. Pages 613-620.

Apratim Das, Mike Drakos, Alex Aravind and Darwin Horning (2019). *Water Governance Network Analysis using Graphlet Mining.* Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), August 27-30, 2019, Vancouver, Canada. Pages 633-640.

Apratim Das, Alex Aravind and Gurbind S. Deo (2019). *Signed Graphlets based Gene Expression Analysis.* Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), November 18-21, 2019, San Diego, USA. Pages 2055-2062.

Abstract

As the world is flooded with data, the demand for mining data for useful purposes is increasing. An effective techniques is to model the data as networks (graphs) and then apply graph mining techniques for analysis. As on date, the algorithms available to count graphlets and orbits for various types of graphs and their generalizations are limited. The thesis aims to fill the gap by presenting a simple and efficient algorithm for 3-node graphlet and orbit counting that is generic enough to work for both undirected and directed graphs. Our algorithm is compared with the state-of-art algorithms and we show that in most cases our algorithm performs better. We demonstrate our algorithm in three case studies related to (i) enzyme and metabolite correlation network in corn, (ii) watershed governance networks, and (iii) patterns exhibited by co-expression networks of healthy and cancerous stomach cells.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the internet and computing devices have become integral part of our everyday life, we are flooded with data. This trend, leading to what is now popularly referred to as the era of big data, has brought new challenges in analyzing the huge amount of data that we face. Their unstructured nature adds complexity to it. In this context, graphs have been used as an effective representation of important elements and their interactions to reveal hidden patterns in the data. Such networks have been extensively studied and applied in many fields such as mathematics, chemistry, molecular biology, ecology, politics, medicine, finance, trade, etc. In computer science, graphs are fundamental data structures used to represent and process data.

1.1 The Backdrop

Motifs are frequently occurring partial sub-graphs used to detect common structural patterns in a network. When interaction networks are modeled as graphs, graph motifs are typically enumerated and their frequencies analyzed to understand the networks. Enumerating generic motifs (all possible sub-graphs) have been extensively studied. However, due to its generality, the applications of graph motifs are limited. Among them, the motifs of triangular graphs received attention in the graph analysis community due to the mathematical properties and their usefulness to various applications [1]. For example, node clustering is one of the important metrics used to study and compare networks.

A popular approach in social networks is predicting various properties of local structure. Many methods rely on the concept of node similarity, which is generally defined in a local sense. These may include hubs, followers, adversaries and intermediaries between the groups. Counting the number of triangles incident on each node is found to be a very useful metric to determine clustering and for node similarity analysis [1]. In this line of interest, a special type motifs of up to 5 nodes, referred to as *graphlets*, were introduced in 2004 and applied to biological networks [2]. Graphlets are small *induced* sub-graphs. A graphlet with n nodes is referred to as n-node graphlet. To study networks, graphlets became popular when the concepts of *orbits* were introduced later in 2007 and used with graphlet counts [3].

The concept of orbits brought a new dimension by including the topological aspect to expose the richness of the roles that nodes play in the network structure and dynamics. *Automorphism* orbits (in mathematical terms *symmetric groups* of nodes) were used to characterize different topological positions of nodes that participate in the network. Using the concept of orbits, a new metric called *graphlet degree vector* was introduced (not to be confused with degree of a node). Graphlet degree vector of a node is a list of the number of times a node participates as each orbit in the graph [4]. With graphlet degree vectors, the dependency between node roles can be encoded in a symmetric matrix, called *graphlet correlation matrix (GCM)*, to describe the network [4].

Most of the applications of graphlet and orbit counts are based on the assumption that the node's local network topology is in some way related to the functionality of the observed node in the network. Since some graphlet structures play functional roles depending on the application, counting frequencies of these graphlets maps the topology of the network as a whole to the functional aspect that we are interested in. Graphlet and orbit frequencies often carry significant information about the network structure in many domains [5]. While graphlet counts are used to generate graphlet frequency distribution, which is useful for network analysis, orbits can further increase the sensitivity of the statistic at nearly the same cost.

Although undirected graphs have been widely used to represent all kinds of networks due to its simplicity and generality, such modelling limits the usefulness and expressive power of the network. In most real world applications, added factors such as the direction of the control and flow of the information from one node to another is very critical. Such information is very useful for understanding the power and influence of the node in the network. When each edge is associated with a direction, the resulting graph is called directed graph. Hence, directed graphs appears to be more useful for many real-life applications. Despite this apparent advantage, compared to undirected graphs, research exploration and applications on directed graphs is very limited. This may be due to the complexities involved with respect to modeling and enumerating sub-graphs for network analysis.

Directed graphs appear to be more useful for many real-life applications. For example, directed graphlet analysis successfully uncovered the directed core-broker-periphery organization of the world trade, predicted the economic attributes of the countries, and identified the relationship between the roles of the countries in the world trade network that were not possible using traditional graph analysis [5]. Due to its added information, the number of unique graphlets and orbits for directed graphs increases drastically as the size of the graphlet increases. This sharp increase in the number of graphlets and orbits, in case of directed networks, is a clear indication that the directed graphlet based analysis is more useful when smaller size of graphlets are considered.

The edges in the directed networks allow us to express asymmetric relationship between nodes. In many domains, some interesting symmetric relationships could exist between nodes. For example, in social networks, people could “like" each other or “dislike" each other [6]. In metabolic networks, nodes could influence to “express" together or “inhibit" each other. More generally, nodes could “attract" each other or “repel" each other. Such relationships can be captured using *signed* edges resulting in signed networks. Recently, to study such networks, *signed graphlets and orbits* were defined and introduced [6].

We now provide further details on graphlets and orbits.

1.2 Graphlets and Orbits

In the context of graphlets, directed graphlets were introduced and applied to network analysis in 2016 [5]. Figure 1.1 lists 3-node graphlets and orbits for undirected, directed, and signed networks. Due to its added information, the number of unique graphlets for directed graphs increase drastically as the size of the graphlet increases. For example, the number of graphlets and orbits of same size increase from undirected to directed and from directed to signed networks (Figure 1.1). That is, for node size 3, undirected has 3 graphlets and 4 orbits, directed has 6 graphlets and 13 orbits, and signed has 9 graphlets and 15 orbits. This is an indication that signed graphlets can exhibit more information about the network than the other two.

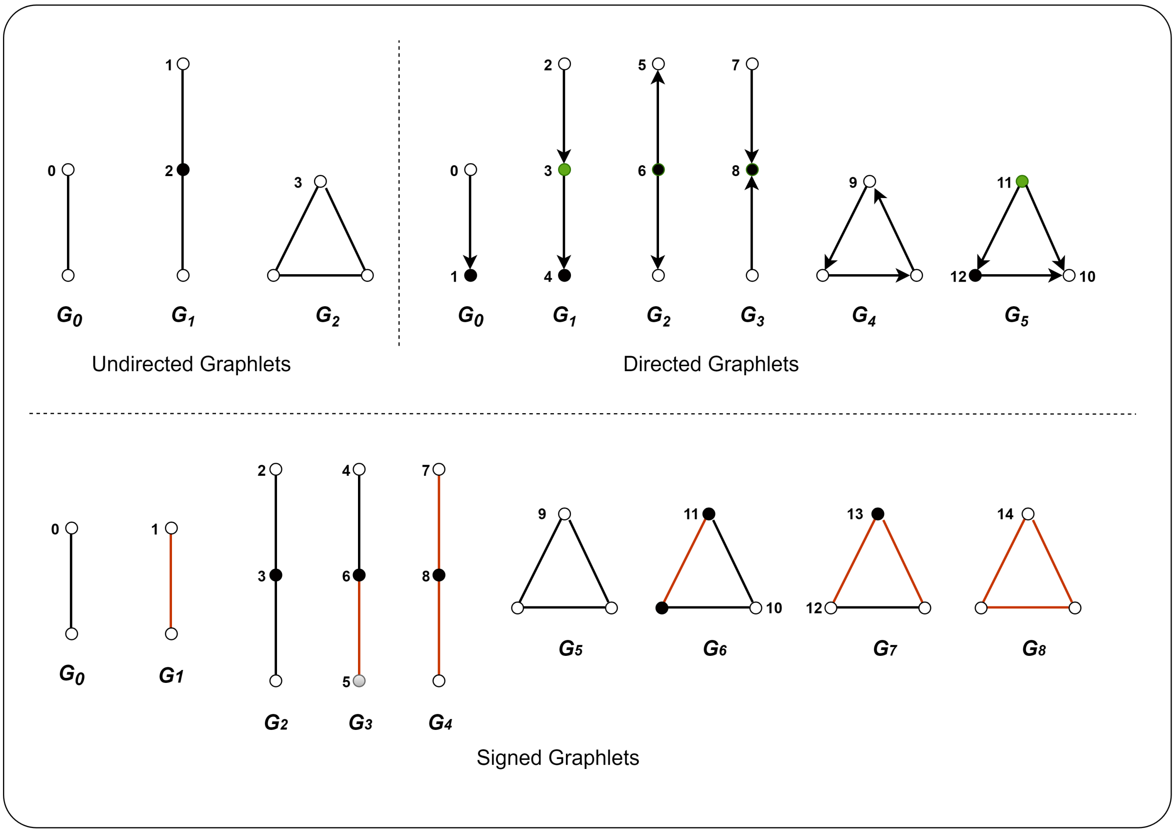


Figure 1.1: Graphlets and Orbits up to 3-nodes

Table 1.1 further illustrates how increase in the size of the graphlet increases the number of non-isomorphic graphlets [7] [8].

Table 1.1: Graphlet Count Growth (Cumulative)

k	k-node graphlets(undirected)	k-node graphlets(directed)
2	1	2
3	3	15
4	9	214
5	30	9,567
6	142	1,540,421
7	965	882,011,563
8	12,082	1,793,355,966,869
9	273,162	13,027,955,038,433,121

This drastic increase in the number of graphlets and orbits in case of directed network is an indication that the directed graphlet based analysis is more useful when smaller size of graphlets are considered. That is, more useful hidden structures could be exposed using undirected graphlets. For example, undirected graphlet analysis successfully uncovered the directed core-broker-periphery organization of the world trade, predicted the economic attributes of the countries, and identified the relationship between the roles of the countries in the world trade network that were not possible using traditional graph analysis [5]. Sarajlić et. al. also demonstrated the power of directed graphlet in analysis of metabolic reactions in the metabolic network of human. From these, it is apparent that directed graphlets have more appeal for practical use than undirected graphlets.

Combining this concept of directed and signed graphlets would give us signed digraphlets. While this has not been considered as part of this research, it should be noted that signed digraphlets can be particularly useful in situations where causal information (direction) and correlation information (sign) are both present.

For the networks of large size, with millions of nodes, enumerating graphlets is a time consuming operation. Particularly, the combinatorial expansion of different types of graphlets is more dramatic when directed and multi-label edges are considered [9]. Also, due to large number of graphlets involved for higher number of nodes, the analysis becomes harder and obscure. The noise introduced by the larger number of graphlets seems to increase. Noise in this context is the statistical repetetion caused due to the counts of orbits being dependent. This brings an important question of how useful the graphlet analysis using 4-node graphlets compare to 3-node graphlets. Focusing on 3-node graphlets may seem trivial since there are only 2 graphlets and 3 orbits for 3-node graphlets. However, if we consider directed graphlets, we have 5 graphlets and 11 orbits. Interestingly, in case of directed graphlets, it is found that 3-node graphlet analysis induces less noise compared to 4-node graphlet analysis [5] [10]. Also, the analysis using 3-node graphlets is expected to be faster than 4-node graphlets, and hence 3-node graphlet analysis may be more preferable.

1.3 Terminology

We start with the terminology used in computing graphlet metrics for given network, say N. Let n be the number of nodes in the network and m be the number of orbits.

- *Signed Network:* Signed network is represented as an adjacency matrix with signs:

$$adj_{i,j} = \begin{cases} 1 & : negative\ edge \\ -1 & : positive\ edge \\ 0 & : not\ adjacent \end{cases}$$

- *Induced sub-graph* - a sub-graph that includes all the edges between its nodes.
- *Graphlet* - a small *induced* connected sub-graph.
- *Graphlet (Automorphism) Orbit* - a *topologically indistinguishable* position in a graphlet.
- *GOV_i:* Graphlet Orbit Vector[1] for node i. GOV_i is a vector of size m and $GOV_i[j]$ contains the number of times the node i participates in orbit j, where $0 \leq j < m$.

[1] Originally referred to as Graphlet Degree Vector (*GDV*) or Graphlet Degree Signature (*GDS*), and we rename it to avoid avoid confusion with usual definition of a node's degree.

- *GOM:* Graphlet Orbit Matrix of the network. *GOM* is an $n \times m$ matrix formed by all graphlet orbit vectors of the nodes in the network.

$$GOM(N) = \begin{bmatrix} G_{0,0} & G_{0,1} & \cdots & G_{0,14} \\ G_{1,0} & G_{0,1} & \cdots & G_{0,14} \\ \vdots & \vdots & \ddots & \vdots \\ G_{n-1,0} & G_{n-1,1} & \cdots & G_{n-1,14} \end{bmatrix}$$

- *GNV$_j$:* Graphlet Node Vector for orbit j. GNV_j is a vector of size n and $GNV_j[i]$ contains the number of times node i participates in orbit j.

- *GOD:* Graphlet Orbit Distribution (GOD) vector. *GOD* is a vector of size m and $GOD[k]$ contains the sum of participation of all nodes in orbit k.

- *GCM:* Graphlet Correlation Matrix (GCM) of the network. It is a symmetric matrix where $GCM[j,k]$ contains the correlation between orbits j and k (i.e., the correlation coefficient is computed from GNV_j and GNV_k).

- *GCD* : Graphlet Correlation Distance (GCD) is a measure of structural distance between two networks [4]. It is taken as the sum of difference of upper right triangles of a pair of GCMs.

$$GCD(G_1,G_2) = \sqrt{\sum_{i=1}^{d} \sum_{j=i+1}^{d} (GCM_{G_1}(i,j) - GCM_{G_2}(i,j))^2}$$

Note that GOV_i and GNV_j, respectively, are the i^{th} row and j^{th} column of GOM.

1.4 Outline of the Thesis

The thesis is organized as follows. Chapter 2 reviews the literature on graphlets. Chapter 3 presents the proposed algorithm for counting 3-nodes graphlets and orbits. In Chapter 4, the proposed algorithm is analyzed for its comparative performance. In Chapter 5 we carry out the graphlet analysis metrics. In Chapter 6, we provide some case studies on the application of graphlets in real world problems. Finally, in Chapter 7 we present a summary and discuss future work.

Chapter 2

Related Works and Literature

Here we briefly review the main works related to graphlet and orbit counting and corresponding metrics. In the context of motif counting, many algorithm have been proposed in the literature. Since graphlets are special type of motifs, technically, any of these algorithms can be used to count graphlets as well. There are many algorithms which are used to count network motifs. However, counting motifs is much slower and inefficient compared to the algorithms which have been specially designed to count graphlets. One of the tools which has been popular for counting graphlets is FANMOD and is one of the slower algorithms for graphlet counting. This software uses existing motif counting algorithms to count graphlets. We now describe the works on undirected graphlet counting, undirected graphlet and orbit counting, directed graphlet and orbit counting and triangle counting.

2.1 Undirected Graphlet Counting

The current state of the art algorithm for counting undirected graphlets is by [11], proposed in 2017. This is a combinatorial algorithm that counts up to 4 node graphlets. Since this is the state of art algorithm, we also compare the performance of our algorithm that counts both 3-nodes graphlets and orbits with the 3-nodes version of the algorithm given in [11] which only counts graphlets for directed graphs.

2.2 Undirected Graphlet and Orbit Counting

Among the algorithms for undirected graphlets and orbit counting, ORCA proposed in [12] is the current state of the art algorithm. However, this algorithm has certain limitations: (i) it is specifically designed to take advantage of larger graphlets, 4 nodes and 5 nodes, using orbital equations derivations; (ii) there is no easy way to reduce the logic to work for 3-node graphlets; and (iii) works only for undirected graphlets. It is also a combinatorial approach fine tuned for 4-5 node graphlets. These limitations make this algorithm not comparable to our algorithm. RAGE (Rapid Graphlet Enumerator) is another simple algorithm designed for undirected graphlet and orbit counts [13] . In general it work's slower than ORCA, but since it consists of multiple algorithm's to count each type of graphlet as efficiently as possible, the logic could be expanded to directed graphlets as well. Also, it is easily parallelizable. We compare our algorithm with RAGE for 3-node graphlets and orbits counting.

2.3 Directed Graphlet and Orbit Counting

The work presented in [10] mentions using directed graphlets for comparison of tissue/disease specific integrated networks using directed graphlet signatures. They resort to using the Naïve approach for counting directed graphlets as a result of lack of other better alternatives. The Naïve brute force approach would iterate over all possible set of vertices and have a complexity of $O(n^k)$ as mentioned previously.

2.4 Triangle Counting

Among the fast triangle counting methods, one of these leverages fast Matrix Multiplication for triangle detection. Given that A is an adjacency matrix which is square, if we compute A^n for an adjacency matrix representation of graph, then a value $A^n[i][j]$ represents number of distinct walks between vertex i to j in graph. In A^3, we get all distinct paths of length 3 between every pair of vertices. A triangle is a cyclic path of length three, i.e. begins and ends at same vertex. So $A^3[i][i]$ represents a triangle beginning and ending with vertex i. Using Strassen's algorithm, this is achievable in $O(V^{2.8074})$ time with V being the number of vertices in the graph, and can be improved further using other algorithms. This solution, however, solves only half the problem. While it does count all the triangles, we still need to count the number of 2-star occurrences. Also, this method counts the graphlet without preserving the counts of orbits. Therefore, matrix multiplication might not be the right approach for counting graphlets in our case.

2.5 Two Graphlet Counting Algorithms

We discuss two exisiting algorithms for graphlet counting.

Algorithm 1 counts, for each node $v \in V$, all triangles adjacent to u: The algorithm iterates over all edges in the graph (lines 2-6). For each edge $e(u,v) \in E$, it counts all triangles that are edge joint with $e(u,v)$ (that is $e(u,v)$ is an edge in the triangle), updating the respected triangle count values of both node u and node v. This edge joint triangle count is obtained using the Merge procedure (line 9), which returns all nodes sharing an edge with both u and v. While NodeArray (which is passed "by ref" via V_{arr}) is not useful for counting triangles, it will be useful for subsequent algorithms calling the Merge procedure. Finally, since each node is connected to two edges in each triangle we count each triangle twice. This over-count is fixed in the final post-processing loop (line 8) [13].

Algorithm 2 (TriadCensus) shows how to count graphlet of size $k = 3$ for each edge. There are four possible graphlets of size $k = 3$ nodes, where only $g3_1$ (that is, triangle patterns) and $g3_2$ (that is, 2-star patterns) are connected graphlets. Lines 5–13 of Algorithm 2 show how to find and count triangles incident to an edge. For any edge $e = (u,v)$, a triangle (u,v,w) exists, if and only if w is connected to both u and v. Let Tri_3 be the set of all nodes that form a triangle with $e = (u,v)$, and let $|Tri_e|$ be the number of such triangles. Then, Tri_e

```
Algorithm TriangleCount ($G$)
    for $v \in V(G)$ do
        for $u \in N(v), v < u$ do
            $Merged \leftarrow Merge(G, v, u, V_{arr})$
            $m_7[v] \leftarrow m_7[v] + |Merged|$
            $m_7[u] \leftarrow m_7[u] + |Merged|$
        end
    end
    for $v \in V(G)$ do
        $m_7[v] \leftarrow m_t[v]/2$
    end
Procedure Merge ($G,v,u,NodeArray$)
    for $w \in N(v)$ do
        $NodeArray[w] \leftarrow 0$
    end
    for $w \in N(u) s.t. w \neq v$ do
        $NodeArray[w] \leftarrow 1$
    end
    for $w \in N(v), w \neq u$ do
        if $NodeArray[w] = 1$ then
            $NodeArray[w] \leftarrow 3$
            $list \leftarrow AppendToListlist, w$
        end
        else
            $NodeArray[w] \leftarrow 2$
        end
    end
    return list
```

Algorithm 1: RAGE (Counting Triangles)

is the set of overlapping nodes in the neighborhoods of u and v; $Tri_e = N(u) \cap N(v)$. Where $N(v)$ are the neighbours of node v.

Note that Algorithm 1 counts each triangle three times (one time for each edge in the triangle), and therefore we divide the total count by 3. Now we need to count 2-star patterns (that is, $g3_2$). For any edge e = (u,v), let $Star_e$ be the set of all nodes that form a 2-star with e and $|Star_e|$ be the number of such star patterns. A 2-star pattern (u,v,w) exists, if and only if w is connected to either u or v but not both. Accordingly, $Star_e = Star_u \cup Star_v$, where $Star_u$ and $Star_v$ are the set of nodes that form a 2-star with e centered at u and v, respectively.

```
Algorithm TriadCensus(G = (V,E))
    Initialize all variables
    for parallel $e = (u,v) \in E$ do
        $Star_u = \emptyset, Star_v = \emptyset, Tri_e = \emptyset$
        for $w \in N(u)$ do
            if $w = v$ then
                continue
            end
            Add w to $Star_u$ and set X(w) = 1
        end
        for $w \in N(v)$ do
            if $w = u$ then
                continue
            end
            if $X(w) = 1$ then
                Add w to $Tri_e$
                Remove w from $Star_u$
                else
                    Add w to $Star_v$
                end
            end
            $f(g3_1, G) += |Tri_e|$
            $f(g3_2, G) += |Star_u| + |Star_v|$
            $f(g3_3, G) += |V| - |N(u) \cup N(v)|$
            for $w \in N(u)$ do
                X(w) = 0
            end
        end
    end
    end parallel
    Aggregate counts from all workers
    $f(g3_1, G) = \frac{1}{3} f(g3_1, G)$
    $f(g3_2, G) = \frac{1}{2} f(g3_2, G)$
    $f(g3_4, G) = \binom{|V|}{3} - f(g3_1, G) - f(g3_2, G) - f(g3_3, G)$
    return $f(G_3, G)$
```

Algorithm 2: Triad census algorithm (Nesreen Ahmed)

Similar to counting triangles, Algorithm 1 counts each 2-star pattern two times (one time for each edge in the 2-star). Thus, we divide the sum of all edges by 2. [11]

2.6 Summary of Exisiting Graphlet Counting Algorithms

We first provide a summary table (Table 2.1) of the comaprison between exisiting graphlet counting algorithms.

Table 2.1: Comparison of Algorithms

Algorithm	Graphlet Size	Orbit Counting	Graph Type
Naïve(Brute)	**Any**	Yes	**All**
RAGE	3,4	Yes	Undirected
Combinatorics	3,4	**No**	Undirected
ORCA	4,5	Yes	Undirected
FANMOD	3-8	Yes	Directed/Undirected

From the above discussion, we list the following observations.

O1: Orbit counts are often critically useful for network analysis in exposing hidden structures related to the dynamics of networks rather than using graphlet counts alone.

O2: Directed graphlets have increased appeal for many real-world applications due to their enhanced expressive power compared to undirected graphlets.

O3: Many real world systems can be effectively modeled as correlation networks and such networks are inherently signed networks.

O4: The number of n-node graphlets and orbits ($n > 1$) increases from undirected to directed, and further increases from directed to signed[1].

O5: The number of graphlets of size up to 4 and the number of orbits in them in undirected networks are equal to the number graphlets of size up to 3 and the number of orbits in them in signed networks.

[1] For up to 3 nodes, there are 3 graphlets and 4 orbits for undirected networks, 6 graphlets and 13 orbits for directed networks, 9 graphlets and 15 orbits for signed networks.

O6: For directed networks, 3-node graphlets are more appealing than graphlets with 4 or more nodes because they can: (i) be computed quicker; (ii) have reduced noise during correlation analysis; and (iii) be rich enough to derive useful results [5].[2]

O7: Most popular graphlet and orbit counting algorithms are designed to work only for undirected graphlets. There appears to be no simple way of trivially modifying them to work for signed graphlets.

O8: Triangle counting is a well-researched problem and many efficient algorithms are reported in the literature for counting triangles in a network.

Based on the above observations, we pose the following questions and hypotheses.

Q1: How to exploit and extend triangular counting to construct an algorithm for computing signed graphlets up to 3-nodes and their orbits?

Q2: Are there applications for signed graphlets and orbits?

H1: Signed graphlets and orbits have increased appeal for many real-world applications due to their expressive power compared to undirected graphlets and orbits.

H2: For signed networks, 3-nodes graphlets and orbits are more appealing[3].

Our contribution in this thesis primarily answers the questions Q1 and Q2 and supports the hypotheses H1 and H2. More specifically, our contributions in this thesis include:

1. An efficient algorithm for enumerating 3-node graphlets and orbits for signed networks. The proposed algorithm is simple and easily parallelizable.
2. A performance comparison of the proposed algorithm with state-of-art algorithms for 3-node graphlets.

[2] Regarding the noise argument for preferring 3-node digraphlets over 4-node, the main motivation was from the analysis done in [5] by Sarajlić et. al. Figure 2 in their paper shows the precision-recall curves for 3-, 4-node digraphs (among others) where the best results were from the 3-node digraph correlation distance. This was done for both induced noise and incompleteness in model networks.

[3] This hypothesis is inspired from the reasons given in observation O6.

3. An illustration of power and use of signed graphlets and orbits by analyzing networks in few case studies.

Chapter 3

Proposed Algorithm

The sufficient background and motivation outlined in the previous chapters makes the algorithm presentation straightforward and relatively easy. However, later in a chapter we will also go through some of the existing graphlet counting algorithms which we have used as benchmarks for our algorithm.

3.1 Basic Idea

The basic idea behind the algorithm is very simple.

Choose the best triangle counting algorithm, and extend it with 2-star counting component to enumerate 3-node graphlets and orbits. The popular algorithms available in the literature to count triangles are given in [1, 14–17]. Among them, the best algorithm is given in [17] and adopted later in [15]. This algorithm is based on a simple and powerful observation that the lowest degree node in each triangle is responsible for making sure the triangle gets counted. Hence any computation related to other vertices of the triangle can be safely ignored without losing the count of that triangle. This simple optimization not only ensure that each triangle in the graph is counted exactly once, but also reduces one third of the computation in the triangle counting otherwise.

With this efficient triangle counting module ready, we now need to add a module for counting 2-stars, which is a straight forward procedure. Similar optimization approach is followed to ensure that a unique structure is counted exactly once. This way, we are able to to avoid multiple counts of these structures. That is the key reason that our algorithm performs better than RAGE [13] and [11].

Another important goal of our algorithm is to make the algorithm generic. As shown in Figure 1.1, there exists only two possible three node graphlet structures for undirected graphlets. In the case of directed and signed graphlets, we have more possible combinations within those structures depending on their orientation. After these two basic structures are identified, then map them to their respective orientation and that could be done in constant time. Since the algorithm allows easy parallelization, in the next section we present the parallelized version of the algorithm as Algorithm 3.

3.2 Algorithm

The algorithm starts with initializing three lists, for storing temp data for 2-star counting, and a list of orbits which contains n rows for each node and k columns – the total number of orbits for the graphlet. This value could change depending on whether the graph is undirected or directed. We then begin the procedure for counting 3-cycles or triangles from line 5. We iterate over every vertex from the graph and for each vertex we permute over all its neighbor pairs (u,w) such that $d_u, d_w > d_v$ and u, w are adjacent, where d_v is the degree of node v. At this state we have identified a 3-cycle structure and increment the orbits of the nodes involved accordingly in the IncrementOrbit function. This concludes the process of counting triangles. Since we are partitioning our iteration on the basis of degree of vertices, we are guaranteed to encounter a 3-cycle only once. We then begin the procedure for counting 2-stars. For every vertex v from the graph and for all its neighbors u, if $v < u$ then, we create a list containing the difference of the set containing lists of neighbors of u and v respectively. This $diffList$ contains the list of all node which are neighbors of u and not neighbors of v. As a result for each w in the $diffList$, v, u, w forms a 2-star centered at u. Since we restrict

$v < u$, we ensure that the 2-star is encountered only once. The orbits of these nodes are then incremented accordingly as per the orbitIncrement function.

```
/* Notation                                                        */
$< x,y >$ : pair of nodes.
$(x,y)$ : edge connecting $x$ and $y$.
/* Initialization                                                  */
INPUT: $G = (V,E)$
OUTPUT: $GOM$ for $G$
$uList = \emptyset, vList = \emptyset, diffList = \emptyset$;
Shared Array: $GOM[\,][\,]$;
Local array: $orbitset$;
for parallel $v \in V$ do
    for each $(< u,w > \in N(v) \wedge (u,w) \in E$ do
        if $(deg[u] > deg[v] \wedge deg[w] > deg[v])$ then
            $orbitset = CalcOrbits(v,u,w)$;
        end
    end
    for $u \in N(v)$ do
        if $v < u$ then
            $uList \leftarrow \{w \mid w \in N(u)\} - \{v\}$;
            $vList \leftarrow \{w \mid w \in N(v)\} - \{u\}$;
            $diffList \leftarrow uList \setminus vList$
            for $w \in diffList \wedge w < v$ do
                $orbitset = CalcOrbits(v,u,w)$;
            end
        end
    end
end
Procedure CalcOrbits(v,u,w)
    $o_v, o_u, o_w = isomorphismMap(v,u,w)$;
    /* computes isomorphic mapping of subgraph(v,u,w) with set of
       all possible 3 node graphlets.                              */
    Increment $GOM[v][o_v]$;
    Increment $GOM[u][o_u]$;
    Increment $GOM[w][o_w]$;
```

Algorithm 3: Graphlet and orbit counting algorithm

Chapter 4

Algorithm Performace Analysis

We first setup the experiment and then provide a performance anaysis of our proposed algorithm.

4.1 The Experimental Setup

Computing Environment: The source code is written in C++ and compiled with flag -O3. Cmake with msbuild were used as the build system. This was run on a 4 core i7-7700 CPU @3.8GHz with hyperthreading (2 threads per core).

Data: We used the publicly available network data from network repository [18] and Stanford Large Network Dataset Collection [19]. These datasets vary from financial networks to social networks to biological and are ordered by the density of the graph for these networks as shown in Table 4.1. While there are many attributes of a network which can affect runtime of our algorithm, we believe that in this case, graph density correlates with our improvement factors.

We conducted three sets of experiments, first one comparing with RAGE shown in Figure 4.1, next one comparing with Triad Census shown in Figure 4.2, and the third one comparing with naïve algorithm shown in Figure 4.3.

Table 4.1: Input Graphs

File	$\|V\|$	$\|E\|$	Density	Dmax	Dmin
MANN	3k	6m	1	3.3k	3.3k
frb100	4k	7.4m	0.92	3.9k	3.6k
C2000	2k	1.8m	0.9	1.8k	1.8k
frb59	1.5k	1m	0.89	1.5k	1.3k
C4000	4k	4m	0.5	2.1k	1.9k
scc	6.8k	4.1m	0.2	6.1k	1
co-papers	434k	16m	2.00E-03	1.2k	1
pokec	1.6m	30.6m	4.60E-05	20.5k	5
ca-IMDB	896k	3.8m	9.40E-06	1.6k	1
dbpedia	590.1k	637.1k	3.66E-06	111k	1
netherlands	2m	2m	9.90E-07	7	1

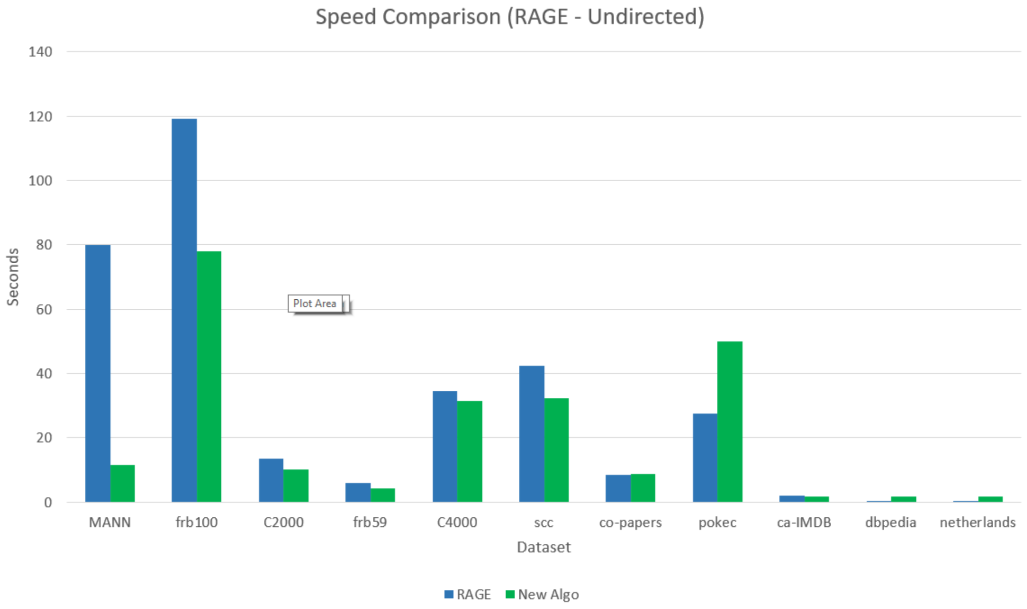


Figure 4.1: Comparing with RAGE

4.2 Performance Analysis

The comparison with RAGE reveals that our algorithm performs better overall. Some of the outliers can be seen on the latter datasets. This is because those are low density graphs with small node degrees. As a result, most of the graphlet counts are weighted towards the number of 2-stars in the graph. Also, higher density graphs naturally have more triangles and our algorithm uses a more efficient triangle counting mechanism, which conforms with

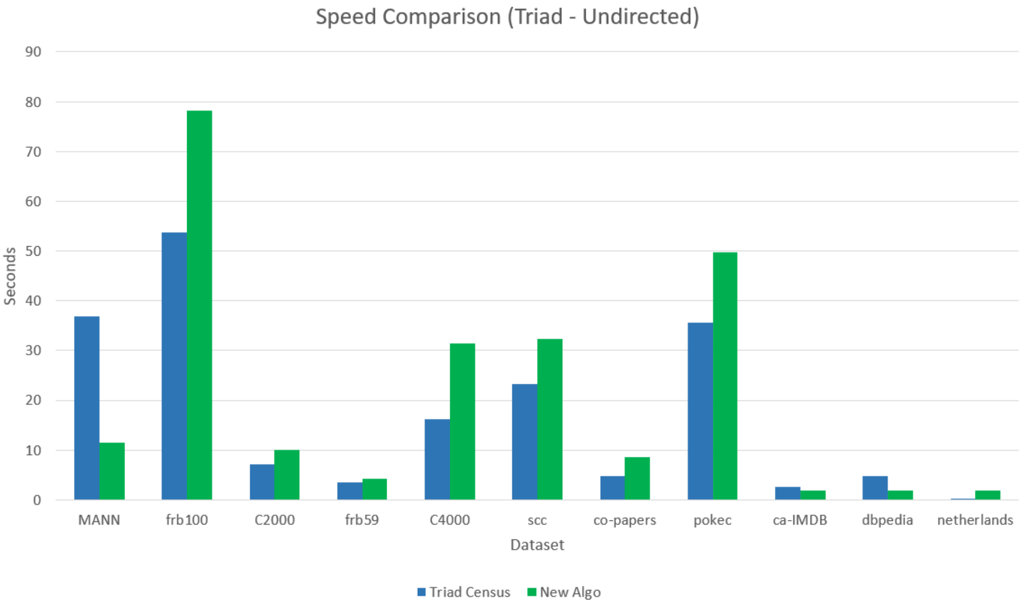


Figure 4.2: Comparing with Triad Census

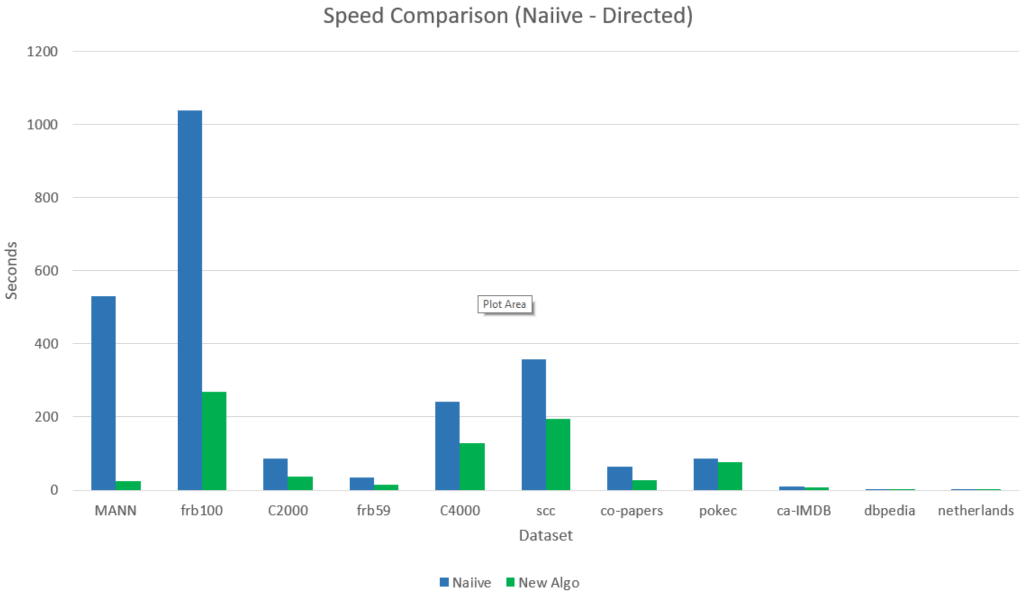


Figure 4.3: Comparing with Naive

our experimental results.

Although Triad Census computes only undirected graphlet counts and our algorithm computes both graphlet and orbit counts, even with added computation, it performs almost comparable to Triad Census.

Along the same lines as the previous comparisons, we have compared orbit counting for directed graphlets with a naïve algorithm which is the only methodology used in literature for the purpose of counting directed graphlets [10]. We can clearly see that our algorithm is a major improvement over the naive implementation and this is more prevalent in higher density graphlets. With these comparisons, we list the speed up over RAGE and Naive algorithms in Table 4.2.

Table 4.2: Speedup Comparison

File	Speed Improvement in Factors	
	vs. RAGE	vs. Naiive (for Directed)
MANN	6.92	22.59
frb100	1.53	3.87
C2000	1.34	2.32
frb59	1.42	2.48
C4000	1.10	1.89
scc	1.31	1.84
co-papers	0.98	2.31
pokec	0.55	1.12
ca-IMDB	1.15	1.49
dbpedia	0.04	0.02
netherlands	0.06	0.54

4.3 Asymptotic Complexity

The space complexity of our algorithm is $O(n)$ which includes the resultant graphlet degree vector storage as well as the lists used to count the 2-stars. The asymptotic time complexity of our algorithm is same of that of RAGE, which is $O(d|E|)$. Here, d is the maximum degree of a node in the graph, E the edge set and $|E|$ is the number of edges. More precisely, the complexity of our algorithm is $(|E|^{\frac{3}{2}} + d.|E|)$. As an additional optimization step, graphs that can be stored as an adjacency matrix in reasonable space are stored as such. This reduces our lookup time for edge existence from $O(log(n))$ to $O(1)$. Note that the naive approach we use compare with our algorithm and has asymptotic complexity $O(n.d^{k-1})$, where k is the graphlet size. So in this case where $k = 3$, the naive runs at $O(n.d^2)$. For constant-time edge queries, the worst-case running time of the triangle counting algorithm is $O(|E|^{3/2})$ [1].

Chapter 5

Graphlet Analysis Metrics

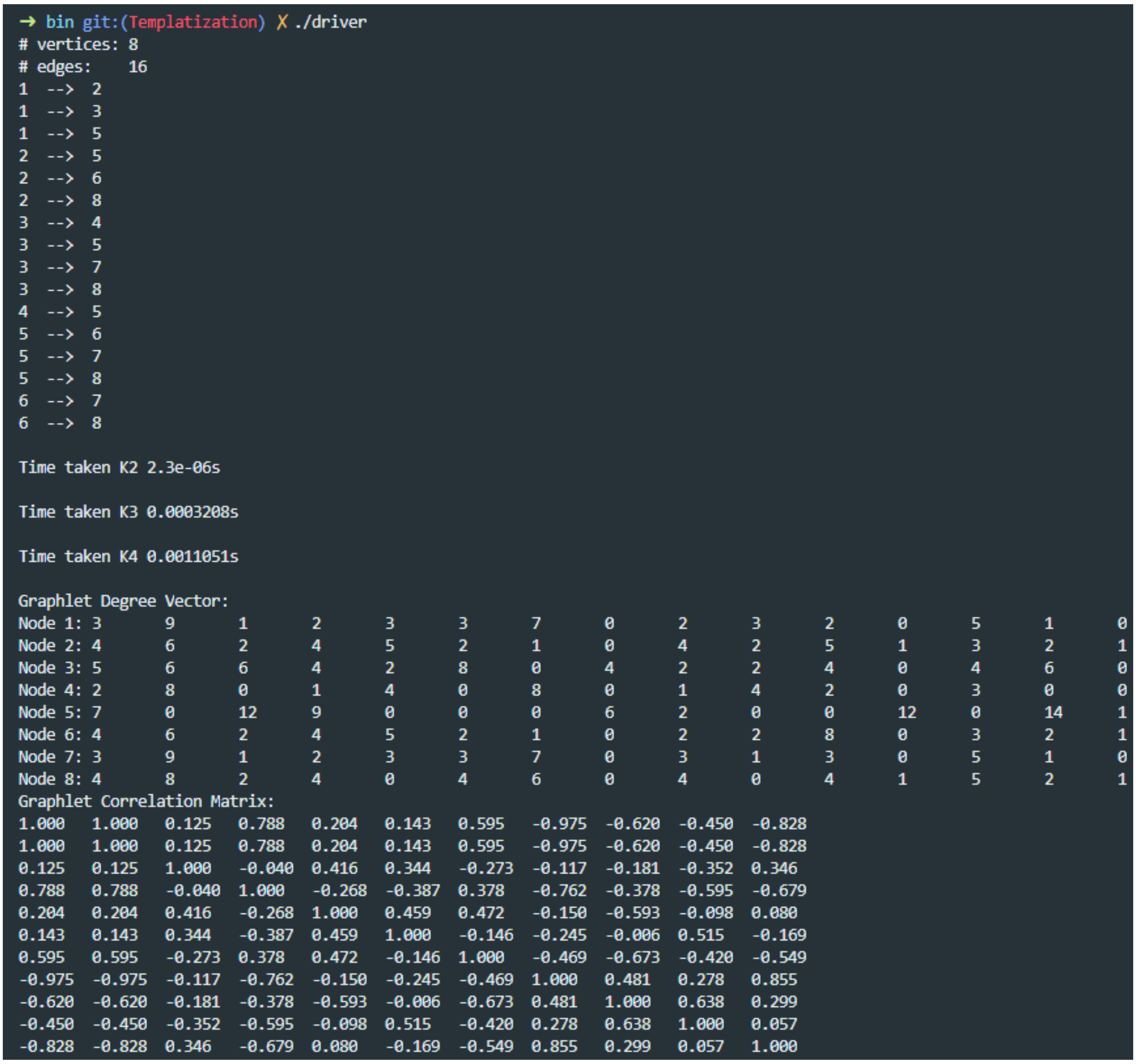
The network analysis essentially has the following main steps [20].

- *Step 1:* Starts with computing graphlet orbit vector for each node in the given network. The computation results in a *GOM*.
- *Step 2:* From *GOM*, graphlet orbit distribution can be drawn and graphlet correlation matrix *GCM* is computed.
- *Step 3:* From *GCM*, corresponding heatmap is drawn for visual interpretation. Using two *GCM*s of two networks, we can get correlation distance between them.

Since graphlet correlation matrix (and hence the heatmap) is of constant size, it eliminates the dependency with the network size. This is an attractive feature of graphlet analysis where a network of any size is reduced to a simple matrix structure of constant size that is convenient for efficient analysis and human interpretations.

By considering up to 4-node graphlets counted using naive approach, Figure 5.1 shows a sample small undirected graph and its resultant correlation matrix and orbit vectors.

We go through some of the existing graphlet counting metrics that have been used. We also extend the same for signed graphlets.



```
➜ bin git:(Templatization) ✗ ./driver
# vertices: 8
# edges:    16
1  -->  2
1  -->  3
1  -->  5
2  -->  5
2  -->  6
2  -->  8
3  -->  4
3  -->  5
3  -->  7
3  -->  8
4  -->  5
5  -->  6
5  -->  7
5  -->  8
6  -->  7
6  -->  8

Time taken K2 2.3e-06s

Time taken K3 0.0003208s

Time taken K4 0.0011051s

Graphlet Degree Vector:
Node 1: 3       9       1       2       3       3       7       0       2       3       2       0       5       1       0
Node 2: 4       6       2       4       5       2       1       0       4       2       5       1       3       2       1
Node 3: 5       6       6       4       2       8       0       4       2       2       4       0       4       6       0
Node 4: 2       8       0       1       4       0       8       0       1       4       2       0       3       0       0
Node 5: 7       0       12      9       0       0       0       6       2       0       0       12      0       14      1
Node 6: 4       6       2       4       5       2       1       0       2       2       8       0       3       2       1
Node 7: 3       9       1       2       3       3       7       0       3       1       3       0       5       1       0
Node 8: 4       8       2       4       0       4       6       0       4       0       4       1       5       2       1
Graphlet Correlation Matrix:
1.000   1.000   0.125   0.788   0.204   0.143   0.595   -0.975  -0.620  -0.450  -0.828
1.000   1.000   0.125   0.788   0.204   0.143   0.595   -0.975  -0.620  -0.450  -0.828
0.125   0.125   1.000   -0.040  0.416   0.344   -0.273  -0.117  -0.181  -0.352  0.346
0.788   0.788   -0.040  1.000   -0.268  -0.387  0.378   -0.762  -0.378  -0.595  -0.679
0.204   0.204   0.416   -0.268  1.000   0.459   0.472   -0.150  -0.593  -0.098  0.080
0.143   0.143   0.344   -0.387  0.459   1.000   -0.146  -0.245  -0.006  0.515   -0.169
0.595   0.595   -0.273  0.378   0.472   -0.146  1.000   -0.469  -0.673  -0.420  -0.549
-0.975  -0.975  -0.117  -0.762  -0.150  -0.245  -0.469  1.000   0.481   0.278   0.855
-0.620  -0.620  -0.181  -0.378  -0.593  -0.006  -0.673  0.481   1.000   0.638   0.299
-0.450  -0.450  -0.352  -0.595  -0.098  0.515   -0.420  0.278   0.638   1.000   0.057
-0.828  -0.828  0.346   -0.679  0.080   -0.169  -0.549  0.855   0.299   0.057   1.000
```

Figure 5.1: Graphlet and Orbit count for sample graph

5.1 Metrics based on Graphlet and Orbit Counts

While graphlets have many applications in data mining and network analysis, their counts are specifically used for identifying correlation of orbit counts for each node within the network. For example, if we consider up to 3-node directed graphlets, each node can participate as one or more of the 13 possible node orbits (Figure 5.1). Once these counts are computed for every node, they are represented as Graphlet Orbit Vectors which is a list of counts of a node's participation in the graph as one of the 13 orbits. We can then identify how these counts are correlated with respect to each other. This leads us to a 13×13, which gets further reduced to 11×11 after identifying and removing redundancies. Graphlet Correlation Matrix summarizing all the data of a network. These correlation matrices can then be used to compare with other correlation matrices of networks to predict how similar their local structure is, or it can be used as a means of measuring change in a dynamic network such as world trade [5]. Others have also used this to study biological networks [10].

In other research, [2] and [3] used Relative Graphlet Frequency (RGF) distance and Graphlet Degree Distribution (GDD)[1] agreement to evaluate the fitness of various network models to real-world networks and to discover a new, well-fitting, geometric random graph model for protein-protein interaction networks, as well as other types of biological networks (such as protein structure networks). GraphCrunch is a software tool for large network analysis and modeling, which uses these graphlet-based network properties [21]. Graphlet Orbit Vectors have also been applied to biological networks to identify clusters of topologically similar nodes in a network and predict biological properties of uncharacterized nodes based on known biological properties of characterized nodes. Specifically, they were applied to protein function prediction [22], cancer gene identification [23], and discovery of pathways underlying certain biological processes.

Automorphism orbit counting requires counting and identifying unique graphlets as a prerequisite. As a result it was assumed that orbit counting is only as computationally expensive as graphlet counting. A newer combinatorial graphlet counting algorithm that counts graphlets via edges is unable to uniquely identify orbits though it gives accurate graphlet counts and is much faster than existing methods [11].

Graphlet frequencies do not uniquely define the network structure, but it has been shown that it can carry significant information about the local network structure in a variety of domains [24]. Since counting orbit comes at an additional cost as compared to counting only graphlets, further analysis on the impact of graphlet frequencies, as compared to orbit counts, would help understand the benefits.

[1] This is different from the GDV/GOV discussed in [2] and [3].

If we consider graphlet orbit vectors and/or graphlet frequencies to be the end result of a successful graphlet counting run-through, the result then has to be useful for analysis.

Yaveroglu et. al. introduced two statistical metrics using the graphlet degree vectors [25]. However, to make their metrics more accurate, they identified relations between the counts of orbits themselves. For example, a 4-node graphlet can be a result of a combination of a 2-node and 3-node graphlet at one node. However, these are also counted and included in the graphlet degree vectors, but, being able to identify and omit these provides more accurate results with less noise. These relations were further extended for directed graphlets for up to 4 nodes by Sarajlić et. al. [5].

The data is then used to identify correlation between each orbit type (*after removing redundant orbits*) using Spearman's Correlation Coefficient. The resultant data is stored as a matrix (the Graphlet Correlation Matrix) which is very useful for understanding characteristics of a network or to compare two networks.

5.2 Graphlet Dependency Equations

Counts of orbits are not independent. As a result, it is useful for us to identify the dependency so as to avoid redundant data as part of our results. Considering only 3- and 4-node graphlets, we also come up with the dependency equations for signed graphlets. As the number of nodes increases, the equations and relations between graphlets turn out to be much harder to identify (as the possible graphlets increase exponentially). The purpose of deriving these dependency equations is to establish the relationship between the counts of the graphlet orbits.

As shown in Figure 5.2, there are two parts to this derivation. The first part identifies which orbits can be obtained as a result of merging two or more smaller graphlets at specified orbits. Once merged, the graph generated (X) along with the vertex at which it was merged (K) will be used to identify the graphlets that are related along with the coefficients. To identify the related graphlets, we consider all permutations of pairs of vertices in X which are not connected. For each of these permutations, we find the graphlet for which X is isomorphic. Once we find this graphlet (G') we then try to find the orbit of the vertex that maps to our focus vertex K, while finding isomorphism. The orbit is them pushed onto an *"orbits"* vector which keeps track of the possible orbits obtained when merging two graphlets. We use the conventional binomial notation when combining orbits.

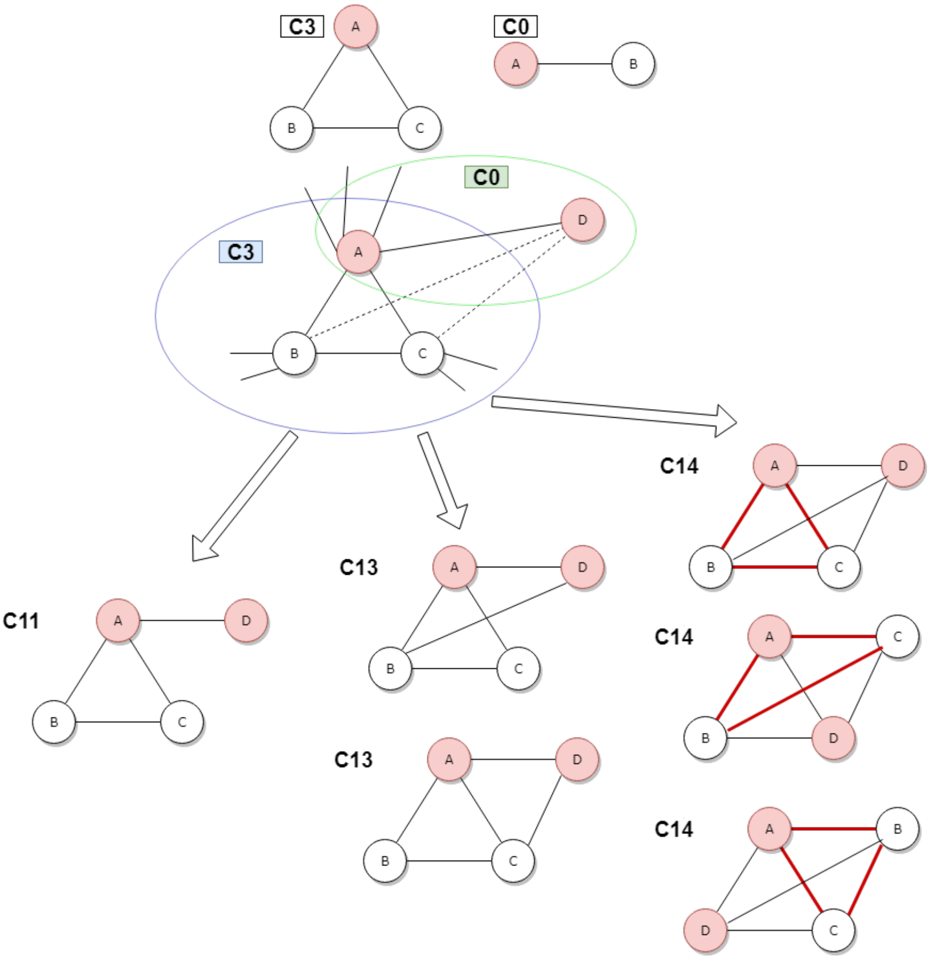


Figure 5.2: Deriving orbital relations

The equations are as follows:

Undirected graphlet relationship $k = 4$ [26]

$$\binom{C_0}{2} = C_2 + \mathbf{C_3}$$

$$\binom{C_2}{1}\binom{C_0-2}{1} = 3C_7 + 2C_{11} + \mathbf{C_{13}}$$
$$\binom{C_1}{1}\binom{C_0-1}{1} = C_5 + 2C_8 + C_{10} + 2\mathbf{C_{12}}$$
$$\binom{C_3}{1}\binom{C_0-2}{1} = C_{11} + 2C_{13} + 3\mathbf{C_{14}}$$
$$\binom{\mathbf{C_0}}{\mathbf{3}} = C_7 + C_{11} + C_{13} + C_{14}$$

Directed graphlet relationship $k = 3$ [5]

$$\binom{C_0}{2} = C_6 + C_{11}$$
$$\binom{C_0}{1}\binom{C_1}{1} = C_3 + C_9 + C_{11}$$
$$\binom{C_1}{2} = C_8 + 2C_{10}$$

Signed graphlet relationship $k = 3$

$$\binom{S_0}{2} = S_2 + S_9 + S_{10}$$
$$\binom{S_0}{1}\binom{S_1}{1} = S_6 + S_{11} + S_{12}$$
$$\binom{S_1}{2} = S_8 + S_{13} + S_{14}$$

The signed graphlets equations were generated as per the existing signed graphlet nomenclature [6]. Extending the equations to k=4 for signed graphlets gives us 29 independent equations.

Signed graphlet relationship $k = 4$

$$\binom{S_2}{1}\binom{S_0-1}{1} = S_{15} + 2S_{41} + S_{42} + S_{52} + S_{61} + 2S_{91} + S_{93} + 2S_{97} + S_{107}$$
$$\binom{S_2}{1}\binom{S_1}{1} = S_{20} + 2S_{43} + S_{46} + S_{58} + S_{72} + 2S_{94} + S_{102} + 2S_{108} + S_{115}$$
$$\binom{S_3}{1}\binom{S_0-1}{1} = 3S_{32} + 2S_{53} + 2S_{58} + S_{92} + S_{95} + S_{103}$$
$$\binom{S_3}{1}\binom{S_1}{1} = S_{35} + 2S_{63} + S_{74} + S_{98} + S_{109} + S_{116}$$

$$\begin{aligned}
\binom{S_4}{1}\binom{S_0-1}{1} &= S_{19}+S_{42}+2S_{45}+S_{55}+S_{68}+S_{93}+2S_{99}+S_{108}+S_{119}\\
\binom{S_4}{1}\binom{S_1}{1} &= S_{24}+S_{44}+S_{48}+S_{65}+S_{79}+S_{105}+S_{111}+S_{118}+S_{124}\\
\binom{S_5}{1}\binom{S_0}{1} &= S_{22}+S_{43}+S_{44}+S_{62}+S_{76}+S_{94}+S_{105}+S_{108}+S_{118}\\
\binom{S_5}{1}\binom{S_1-1}{1} &= S_{28}+2S_{47}+S_{49}+S_{73}+S_{86}+2S_{100}+S_{112}+2S_{120}+S_{125}\\
\binom{S_6}{1}\binom{S_0-1}{1} &= 2S_{35}+2S_{56}+2S_{59}+S_{63}+S_{74}+S_{96}+S_{101}+S_{106}+S_{113}\\
\binom{S_6}{1}\binom{S_1}{1} &= 2S_{38}+S_{70}+2S_{77}+S_{81}+2S_{87}+S_{110}+S_{119}+S_{121}+S_{126}\\
\binom{S_7}{1}\binom{S_0}{1} &= S_{27}+S_{46}+S_{48}+S_{69}+S_{83}+S_{102}+S_{111}+S_{115}+S_{124}\\
\binom{S_7}{1}\binom{S_0-1}{1} &= S_{30}+S_{49}+2S_{50}+S_{80}+S_{89}+S_{112}+2S_{122}+S_{125}+2S_{128}\\
\binom{S_8}{1}\binom{S_0}{1} &= S_{38}+S_{70}+S_{81}+S_{104}+S_{114}+S_{123}\\
\binom{S_8}{1}\binom{S_1-2}{1} &= 3S_{40}+2S_{84}+2S_{90}+S_{117}+S_{127}+S_{129}\\
\binom{S_9}{1}\binom{S_0-2}{1} &= S_{53}+2S_{92}+S_{95}+3S_{130}+2S_{131}+S_{133}\\
\binom{S_9}{1}\binom{S_1}{1} &= S_{56}+S_{96}+S_{110}+S_{132}+S_{134}+S_{137}\\
\binom{S_{10}}{1}\binom{S_0-2}{1} &= S_{59}+S_{95}+2S_{103}+S_{131}+2S_{133}+3S_{139}\\
\binom{S_{10}}{1}\binom{S_1}{1} &= S_{66}+S_{106}+S_{113}+S_{136}+S_{141}+S_{144}\\
\binom{S_{11}}{1}\binom{S_0-1}{1} &= S_{63}+S_{96}+2S_{98}+S_{106}+S_{109}+2S_{132}+S_{134}+2S_{136}+S_{141}\\
\binom{S_{11}}{1}\binom{S_1-1}{1} &= S_{70}+2S_{104}+S_{110}+S_{114}+S_{121}+S_{135}+2S_{140}+S_{142}+S_{145}\\
\binom{S_{12}}{1}\binom{S_0-1}{1} &= S_{74}+S_{101}+S_{109}+S_{113}+2S_{116}+S_{134}+S_{141}+2S_{137}+2S_{144}\\
\binom{S_{12}}{1}\binom{S_1-1}{1} &= S_{81}+S_{112}+S_{119}+2S_{123}+S_{126}+S_{142}+2S_{143}+S_{145}+2S_{147}\\
\binom{S_{13}}{1}\binom{S_0}{1} &= S_{77}+S_{110}+S_{119}+S_{135}+S_{142}+S_{143}\\
\binom{S_{13}}{1}\binom{S_1-2}{1} &= S_{84}+2S_{117}+S_{127}+3S_{138}+2S_{146}+S_{148}
\end{aligned}$$

$$\binom{S_{14}}{1}\binom{S_0}{1} = S_{87}+S_{123}+S_{126}+S_{140}+S_{145}+S_{147}$$
$$\binom{S_{14}}{1}\binom{S_1-2}{1} = S_{90}+S_{127}+2S_{129}+S_{146}+2S_{148}+3S_{149}$$

The following equations can be derived from above

$$\binom{S_0}{3} = \binom{S_3}{1}\binom{S_0-2}{1}+\binom{S_9}{1}\binom{S_0-2}{1}+\binom{S_{10}}{1}\binom{S_0-1}{1}$$
$$\binom{S_1}{3} = \binom{S_8}{1}\binom{S_1-2}{1}+\binom{S_{13}}{1}\binom{S_1-2}{1}+\binom{S_{14}}{1}\binom{S_1-2}{1}$$

Chapter 6

Applications

With the abundance of data, the demand for mining data to gain insights is increasing. One effective technique to deal with the problem is to model the data as networks (graphs) and then apply graph mining techniques to uncover useful patterns. We employ the graphlet-based analysis due to its power in exposing hidden structure and interaction within the networks. Here we take up three case studies relating to (i) metabolite interaction network, (ii) water governamce network and (iii) microarray in gene expression.

6.1 Metabolite Network

Consider intra-cellular networks such as metabolic networks where enzymes and substrates form nodes and possible interaction between them form the edges. In these networks, the interactions between nodes (typically catalyzed by enzymes that act upon substrates) occur within the cell to grow, survive, and reproduce.

6.1.1 Graphlet Analysis with Random Networks

We describe the computation of the metrics listed in Section 1.3 using two synthetic signed networks of different sizes, network density, and proportion of negative edges in them. We will also illustrate the application of signed graphlet analysis for a real network (enzyme metabolic interaction network in corn [27]) from biological domain. The purpose of this section is to show the computational aspects of the analysis irrespective of the particular application domain. Specifically, we expose the difference in the graphlet orbit distribution and corresponding heatmaps representing their correlation matrices for two random networks.

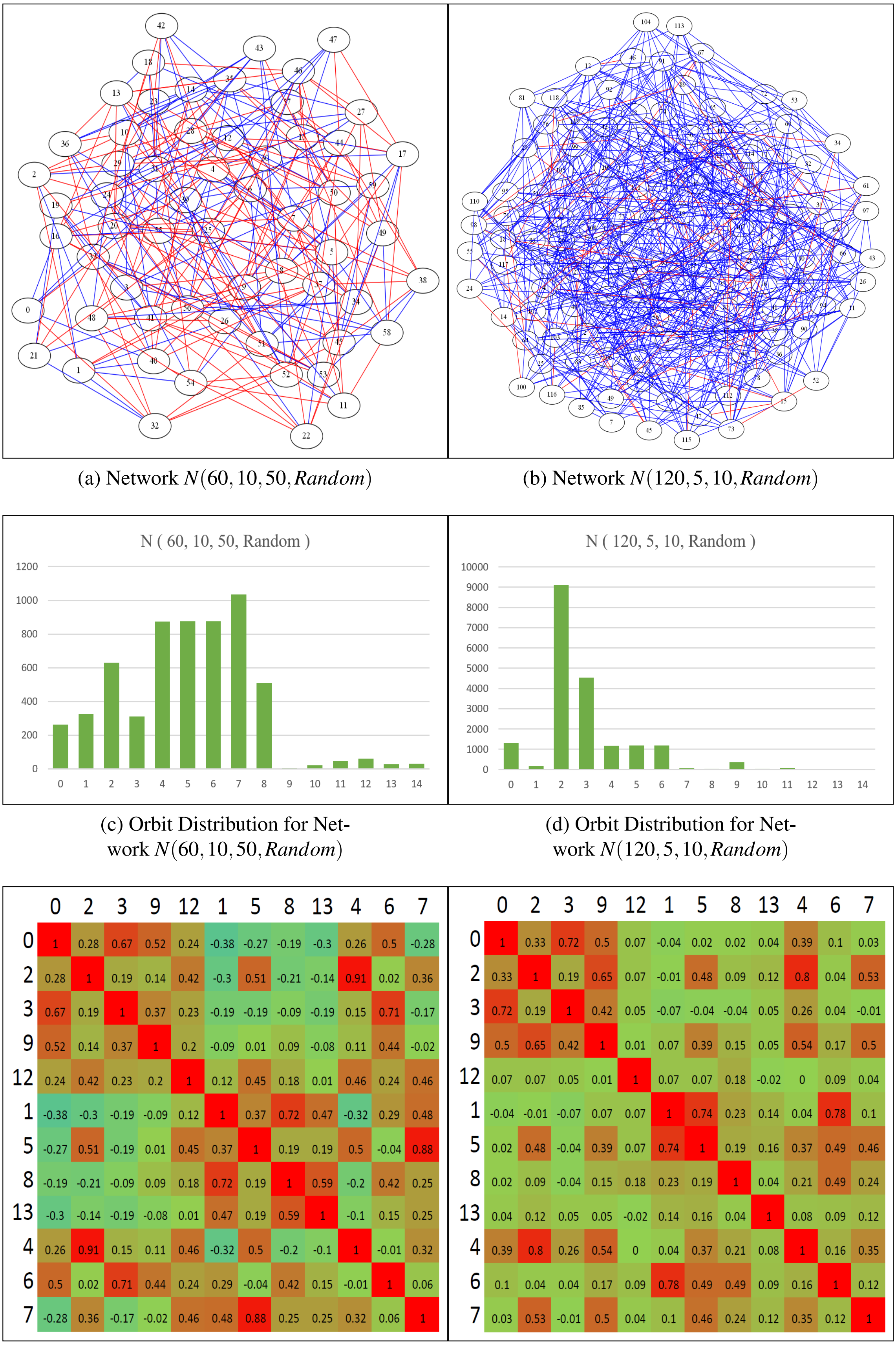
Inspired by biological networks that have very low proportion of negative edges, we modeled one network with a low percentage of negative edges (10%). The other network has equal percentage of positive and negative edges. We also chose low density networks (10% and 5% edges). The first network has 60 nodes (the enzyme metabolic network example has 56 nodes) and the second network has 120 nodes.

We label the networks as *N(n,d,s,t)*, where *n* is the number of nodes, *d* is the edge density, *s* is the percentage of negative edges, and *t* type of network (Real or Random). The blue edges are positive edges and red edges are negative edges. These networks are shown in Figure 6.1(a) and Figure 6.1(b). One notable difference in edge distribution with the enzyme metabolic network is that here the edges are dispersed whereas in the enzyme metabolic network the negative edges originate mainly from one node.

Using the algorithm presented in Section 3.2, we computed graphlet orbit vectors. Based on the orbit vectors, we have drawn the orbit distribution for the two networks, shown in Figure 6.1(c) and Figure 6.1(d). From graphlet orbit matrix, graphlet correlation matrix for both networks are computed. Their heatmap is displayed in Figure 6.1(e) and Figure 6.1(f).

6.1.2 Orbits Optimization

Counts of orbits are not independent of each other as some counts can be derived from the counts of other orbits. This can be resolved based on orbit dependency equations. Based on the dependent equations, we identified orbits 10, 11 and 14 as redundant and therefore do not include them in the final *GCM* computations.



(a) Network $N(60, 10, 50, Random)$

(b) Network $N(120, 5, 10, Random)$

(c) Orbit Distribution for Network $N(60, 10, 50, Random)$

(d) Orbit Distribution for Network $N(120, 5, 10, Random)$

	0	2	3	9	12	1	5	8	13	4	6	7
0	1	0.28	0.67	0.52	0.24	-0.38	-0.27	-0.19	-0.3	0.26	0.5	-0.28
2	0.28	1	0.19	0.14	0.42	-0.3	0.51	-0.21	-0.14	0.91	0.02	0.36
3	0.67	0.19	1	0.37	0.23	-0.19	-0.19	-0.09	-0.19	0.15	0.71	-0.17
9	0.52	0.14	0.37	1	0.2	-0.09	0.01	0.09	-0.08	0.11	0.44	-0.02
12	0.24	0.42	0.23	0.2	1	0.12	0.45	0.18	0.01	0.46	0.24	0.46
1	-0.38	-0.3	-0.19	-0.09	0.12	1	0.37	0.72	0.47	-0.32	0.29	0.48
5	-0.27	0.51	-0.19	0.01	0.45	0.37	1	0.19	0.19	0.5	-0.04	0.88
8	-0.19	-0.21	-0.09	0.09	0.18	0.72	0.19	1	0.59	-0.2	0.42	0.25
13	-0.3	-0.14	-0.19	-0.08	0.01	0.47	0.19	0.59	1	-0.1	0.15	0.25
4	0.26	0.91	0.15	0.11	0.46	-0.32	0.5	-0.2	-0.1	1	-0.01	0.32
6	0.5	0.02	0.71	0.44	0.24	0.29	-0.04	0.42	0.15	-0.01	1	0.06
7	-0.28	0.36	-0.17	-0.02	0.46	0.48	0.88	0.25	0.25	0.32	0.06	1

(e) Heatmap for Network $N(60, 10, 50, Random)$

	0	2	3	9	12	1	5	8	13	4	6	7
0	1	0.33	0.72	0.5	0.07	-0.04	0.02	0.02	0.04	0.39	0.1	0.03
2	0.33	1	0.19	0.65	0.07	-0.01	0.48	0.09	0.12	0.8	0.04	0.53
3	0.72	0.19	1	0.42	0.05	-0.07	-0.04	-0.04	0.05	0.26	0.04	-0.01
9	0.5	0.65	0.42	1	0.01	0.07	0.39	0.15	0.05	0.54	0.17	0.5
12	0.07	0.07	0.05	0.01	1	0.07	0.07	0.18	-0.02	0	0.09	0.04
1	-0.04	-0.01	-0.07	0.07	0.07	1	0.74	0.23	0.14	0.04	0.78	0.1
5	0.02	0.48	-0.04	0.39	0.07	0.74	1	0.19	0.16	0.37	0.49	0.46
8	0.02	0.09	-0.04	0.15	0.18	0.23	0.19	1	0.04	0.21	0.49	0.24
13	0.04	0.12	0.05	0.05	-0.02	0.14	0.16	0.04	1	0.08	0.09	0.12
4	0.39	0.8	0.26	0.54	0	0.04	0.37	0.21	0.08	1	0.16	0.35
6	0.1	0.04	0.04	0.17	0.09	0.78	0.49	0.49	0.09	0.16	1	0.12
7	0.03	0.53	-0.01	0.5	0.04	0.1	0.46	0.24	0.12	0.35	0.12	1

(f) Heatmap for Network $N(120, 5, 10, Random)$

Figure 6.1: Random Networks

6.1.3 Matrix Rearrangement

Heatmaps are used to display patterns that can be interpreted as network characteristics. For that purpose, the rows and columns have been rearranged to enhance patterns in the heatmaps. When comparing two networks in terms of heatmaps or *GCM*s, the rearrangement preserves the partial order of orbits in both matrices.

6.1.4 Observations in Random Networks

From Figure 6.1(c), Figure 6.1(d), Figure 6.1(e), and Figure 6.1(f), we list the following observations. Note that the range of counts in Figure 6.1(c) is 0 to 1200, whereas the range for Figure 6.1(d) is 0 to 10,000 (10 times higher).

- Both random networks *N(60, 10, 50, Random)* and *N(120, 5, 10, Random)* have low graphlet counts for signed graphlets $G5$-$G8$ which includes orbits 9-14. This is because of the sparse nature of the random network discouraging clique formations probabilistically. However, depending on the parameters of generating random networks, this can be different.

- Counts of orbits with positive edges (0,2,3,4,6,9) are much higher than their graphlet counterparts with negative edges for network 6.1(b).

- Orbit 9 is a 3-cycle with all positive edges and therefore has the highest participation compared to other 3-cycles in the network.

- The distribution of graphlet counts in network 6.1(a) in comparison with the counts in network 6.1(b) is much more uniform amongst the graphlets with more positive versus negative edges. Clique participation is still low compared to network 6.1(b). Since network 6.1(b) has twice the density as the network 6.1(a), we see higher participation in orbits 9-14 compared to others.

- With the two points above, we can say that networks that have a high positive to negative edge ratio are likely to see higher orbit participation in some orbits than others. Also, sparse network with a more uniform degree distribution will lead to low counts of triangle graphlet structures $G5$-$G8$ and high counts of two-star graphlet structures $G2$-$G3$.
- By generating the *GCM*s for the two graphs, we can see patterns that are distinct from one another. We calculate the graphlet correlation distance to quantify the structural difference between the networks (Table 6.2).

6.1.5 Metabolite Interaction in Corn

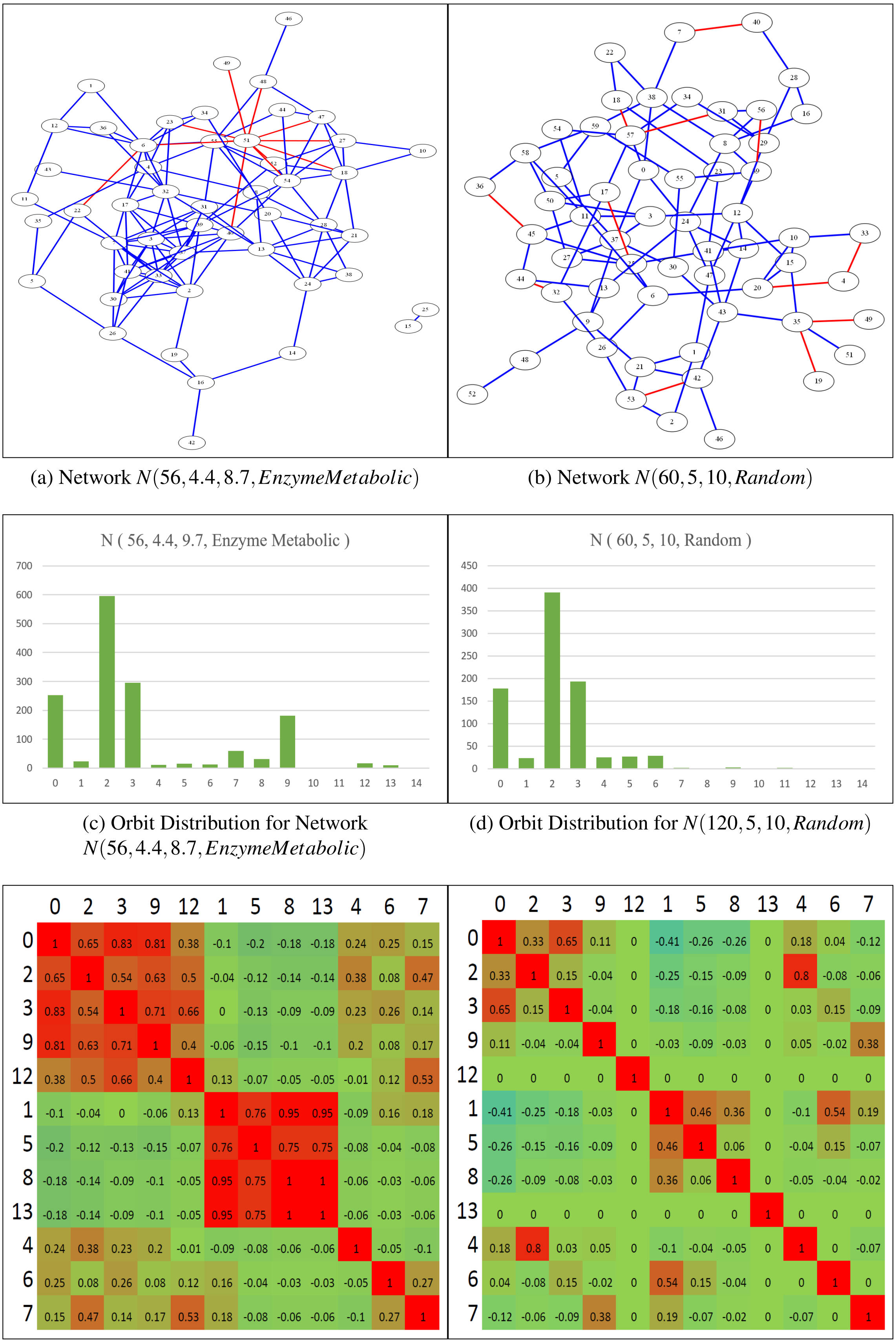
To demonstrate the applications of signed graphlets in real networks, we chose enzyme and metabolite interaction in corn [27]. We constructed the signed network from the correlation matrix of the enzyme metabolic interaction network given in the appendix of the work by Toubiana et. al. [27].

Positive correlation between two nodes is converted to positive edge between those two nodes, and negative correlation is converted to negative edge. The resulting signed network has 56 nodes, 4.4% edge density, and 8.7% negative edges.

In addition to analyzing the enzyme metabolic network, we want to compare it with a synthetic random network. For that we generated a random network has similarity (60 nodes, 5% edge density, and 10% negative edges) to the enzyme metabolic network. These two networks are shown in Figure 6.2(a) and Figure 6.2(b). The orbit distributions of the networks are given in Figure 6.2(c) and Figure 6.2(d).

6.1.6 Modularity Index and Correlation Distance

Modularity is a network measure that quantifies how a given network can be partitioned into communities or clusters [28]. It is defined as the portion of the edge connections within



(a) Network $N(56, 4.4, 8.7, EnzymeMetabolic)$

(b) Network $N(60, 5, 10, Random)$

(c) Orbit Distribution for Network $N(56, 4.4, 8.7, EnzymeMetabolic)$

(d) Orbit Distribution for $N(120, 5, 10, Random)$

	0	2	3	9	12	1	5	8	13	4	6	7
0	1	0.65	0.83	0.81	0.38	-0.1	-0.2	-0.18	-0.18	0.24	0.25	0.15
2	0.65	1	0.54	0.63	0.5	-0.04	-0.12	-0.14	-0.14	0.38	0.08	0.47
3	0.83	0.54	1	0.71	0.66	0	-0.13	-0.09	-0.09	0.23	0.26	0.14
9	0.81	0.63	0.71	1	0.4	-0.06	-0.15	-0.1	-0.1	0.2	0.08	0.17
12	0.38	0.5	0.66	0.4	1	0.13	-0.07	-0.05	-0.05	-0.01	0.12	0.53
1	-0.1	-0.04	0	-0.06	0.13	1	0.76	0.95	0.95	-0.09	0.16	0.18
5	-0.2	-0.12	-0.13	-0.15	-0.07	0.76	1	0.75	0.75	-0.08	-0.04	-0.08
8	-0.18	-0.14	-0.09	-0.1	-0.05	0.95	0.75	1	1	-0.06	-0.03	-0.06
13	-0.18	-0.14	-0.09	-0.1	-0.05	0.95	0.75	1	1	-0.06	-0.03	-0.06
4	0.24	0.38	0.23	0.2	-0.01	-0.09	-0.08	-0.06	-0.06	1	-0.05	-0.1
6	0.25	0.08	0.26	0.08	0.12	0.16	-0.04	-0.03	-0.03	-0.05	1	0.27
7	0.15	0.47	0.14	0.17	0.53	0.18	-0.08	-0.06	-0.06	-0.1	0.27	1

(e) Heatmap for Network $N(56, 4.4, 8.7, EnzymeMetabolic)$

	0	2	3	9	12	1	5	8	13	4	6	7
0	1	0.33	0.65	0.11	0	-0.41	-0.26	-0.26	0	0.18	0.04	-0.12
2	0.33	1	0.15	-0.04	0	-0.25	-0.15	-0.09	0	0.8	-0.08	-0.06
3	0.65	0.15	1	-0.04	0	-0.18	-0.16	-0.08	0	0.03	0.15	-0.09
9	0.11	-0.04	-0.04	1	0	-0.03	-0.09	-0.03	0	0.05	-0.02	0.38
12	0	0	0	0	1	0	0	0	0	0	0	0
1	-0.41	-0.25	-0.18	-0.03	0	1	0.46	0.36	0	-0.1	0.54	0.19
5	-0.26	-0.15	-0.16	-0.09	0	0.46	1	0.06	0	-0.04	0.15	-0.07
8	-0.26	-0.09	-0.08	-0.03	0	0.36	0.06	1	0	-0.05	-0.04	-0.02
13	0	0	0	0	0	0	0	0	1	0	0	0
4	0.18	0.8	0.03	0.05	0	-0.1	-0.04	-0.05	0	1	0	-0.07
6	0.04	-0.08	0.15	-0.02	0	0.54	0.15	-0.04	0	0	1	0
7	-0.12	-0.06	-0.09	0.38	0	0.19	-0.07	-0.02	0	-0.07	0	1

(f) Heatmap for $N(60, 5, 10, Random)$

Figure 6.2: Real Network vs. Random Network

the same cluster minus the expected portion if the connections were distributed randomly. Correlation distance is a measure of similarity between networks.

We computed modularity indices for all 4 networks and the correlation distances between them. More precisely, for correlation distance, we computed euclidean distance between the correlation matrices. These metrics both modularity indices and correlation distances are given in Table 6.1 and Table 6.2.

Table 6.1: Modularity Indices for the Networks

EN	N_1	N_2	N_3
0.513	0.548	0.259	0.253

$EN \ - \ N(56, 4.4, 8.7, EnzymeMetabolic)$

Table 6.2: Graphlet Correlation Distance

	N_1	N_2	N_3
EN	9.47	12.84	13.29
N_1		8.08	5.96
N_2			8.83

$N_1 \ - \ N(60, 5, 10, Random)$
$N_2 \ - \ N(60, 10, 50, Random)$
$N_3 \ - \ N(120, 5, 10, Random).$

6.1.7 Observations on Metabolite Interaction

From Figure 6.2(c), Figure 6.2(d), Figure 6.2(e), and Figure 6.2(f), we list the following observations.

- The orbit distribution between network *N(56, 4.4, 8.7, Enzyme Metabolic)* and the random network *N(60, 5, 10, Random)* is fairly similar.

- Some orbits do not participate in the network such as orbits (10,11 and 14) for the enzyme metabolite network and orbits (12, 13 and 14) for *N(60, 5, 10, Random)*. This is to be expected since the sparseness of the network along with the low negative edge ratio results in some of the graphlet structures not being expressed in the network.

- In terms of dissimilarities, we see that there is more participation of orbits 9, 12 and 13 for the network *N(56, 4.4, 8.7, Enzyme Metabolic)* versus the network *N(60, 5, 10, Random)*. This is because in network *N(56, 4.4, 8.7, Enzyme Metabolic)*, the negative edges mostly originate from the same node and that contribute to higher clique participation. Also, the network *N(56, 4.4, 8.7, Enzyme Metabolic)* has more cliques compared to *N(60, 5, 10, Random)*.

- There are two groups of orbits (0,2,3,9,12) and (1,5,8,13) in *GCM* of the network *N(56, 4.4, 8.7, Enzyme Metabolic)* that exhibit high correlation amongst orbits in the same groups and low to no correlation between orbits from other group. This grouping is less pronounced in the random network *N(60, 5, 10, Random)*.

- The above difference in heatmaps of the network *N(56, 4.4, 8.7, Enzyme Metabolic)* and the network *N(60, 5, 10, Random)* becomes more apparent, even though the only major difference in the two networks is how the negative edges are distributed.

- As a result of the comparability power demonstrated above, we can extend this analysis for metabolite interactions in other species and see how they differ.

- The modularity index (Table 6.1) provides a measure of partitionability of a network. This is another metric with which we can assert the structural similarity of the network. While EN,N_1, and N_2, N_3 have their structural similarity on the basis of the index, we can see more granular differences being reflected in the *GCM*'s for each of the networks.

- From Table 6.2 of Euclidean distance between the networks, we can see that *GCD* of the network (*EN*) is lowest (indicating closest) with the first random network N_1 compared to the others. Obviously, the N_1 is more closer to N_3 and it is reflected in the smallest distance between them.

6.2 Water Governance Network Analysis

Many factors can accelerate the scarcity and quality of water available for basic necessity such as drinking, irrigation, manufacturing, domestic use, etc. Growing population, urbanization, increased sophistication in life, global warming, and climate change are some of these factors and their influence is expected to worsen the scarcity and quality of water in coming decades. Therefore, a sustainable water governance is essential across the world to face this challenge [29].

The exponential growth in the use of social network analysis in the field of environmental governance has been primarily driven by two key factors. There is growing recognition that the governance of socioecological systems is becoming increasingly complex as climate change drives an increase in non-stationarity, while continued failures in past and current governance approaches have spurred on a general ‘rethinking’ of governance approaches and subsequently, the methodologies used to investigate them.

Social network analysis (SNA), as applied to water governance, currently uses common heuristics (e.g., degree centrality) to identify and describe network properties (e.g., structure and node roles). Network statistics are used to approximate network structure for comparisons between either two or more networks, or between networks and an identified typology [30] [31]. This section investigates the use of different network analysis techniques that have, as yet, not been applied to water governance communication networks. Based on graphlet theory, these techniques perform similar types of analysis, but use a different approach, including both graphlet orbit vectors and graphlet correlation matrices to reveal the underlying network structure.

This section compares the two network analysis approaches by applying them to two complex watershed planning net- works. The networks are first analyzed using standard heuristics (i.e., centrality, betweenness, and clustering) and then graphlet theory. The purpose is to: (1) obtain a deeper understanding of the network structure and extract information that is not normally revealed by standard heuristics, (2) quantify structural information in a way that allows network comparisons to be easily made, and (3) identify an approach for future socioecological analysis based upon findings in steps (1) and (2).

This study will show that the use of graphlet theory analysis gives insight into both structure and nodal characteristics and that network analysis using graphlet theory provides a sophisticated approach for analyzing the complex networks that constitute socioecological systems.

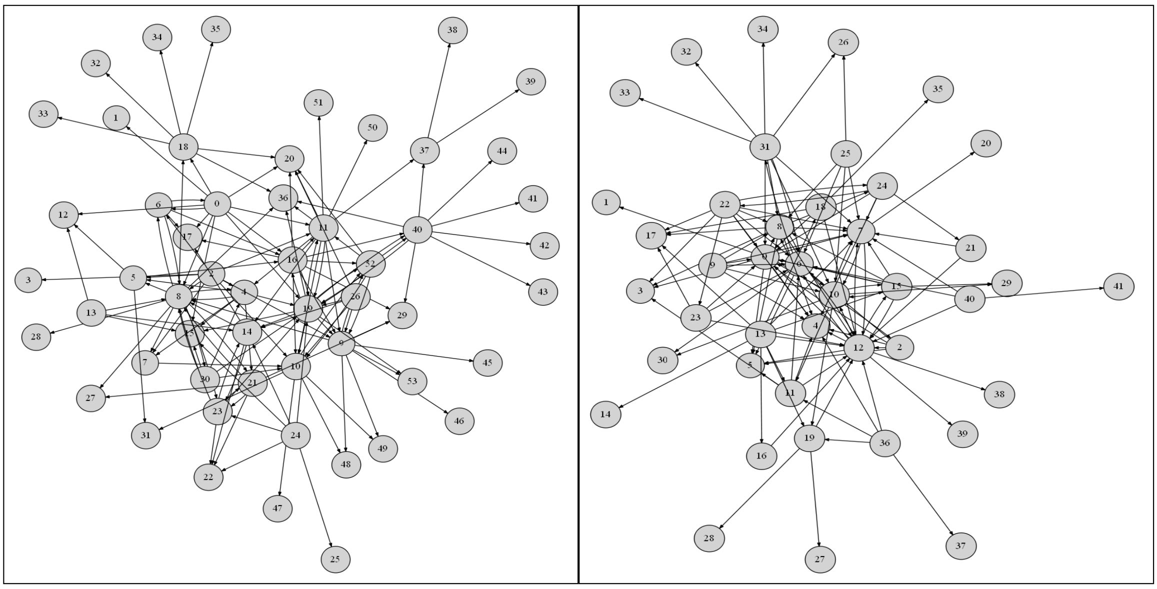
6.2.1 Watershed Governance Networks

The two case study watersheds, Similkameen Valley Watershed (SVW) and the Kettle River Watershed (KRW), were selected based upon similar regulatory, environmental, and socio-economic contexts, and strong parallels in the underlying water-related drivers for initiating watershed planning (i.e., increasing demand, changing supply, and conflicting views on legislative and regulating roles in common pool resource management in British Columbia).

The Similkameen Valley Watershed Plan started as part of Similkameen Valley Sustainability Strategy (Sustainability Strategy). A cross-section of SVPS panel members were selected following a recruitment process that included valley-wide advertising and requests. Expertise of the panel members covered a broad range, including agriculture, arts/culture, business, education, environment/archeology, tourism, and science.

KRW watershed residents' concerns about water supply (e.g., quantity and quality), adequate quantity for ecosystems (e.g., flow for fish), and the uncertainties associated with the impact of a warming climate [32], led to the regional government (Regional District of Kootenay Boundary) applying, successfully, for a federal government grant to develop a Watershed Management Plan (WMP). The WMP consisted of two phases: (1) the Technical Assessment intended to summarize the "State of the Kettle River Water-shed", including any gaps in information, and (2) the Watershed Management Plan, which identified the planning goals, actions, and policy needed to maintain the health of the watershed over the long term [32].

An initial Kettle River Watershed Committee, made up of government representatives and appointed stakeholders, acted as the Steering Committee for the Plan. The Steering Committee then appointed the Technical Advisory Committee (TAC) and the Stakeholder Advisory Committee (SAC). The SAC was designed to include a balanced representation from the various sectors in the watershed (i.e., government, industry, tourism-recreation), with the primary purpose of advising the Steering Committee. The TAC consisted of various government representatives, with some representation from First Nations and industry. The networks of Similikameen and Kettle River are given in Figure 6.3.



(a) Similkameen (b) Kettle

Figure 6.3: Watershed Network Sociograph

6.2.2 Watershed using Social Network Analysis

The existence of social networks is an important aspect of effective multi-stakeholder natural-resource decision-making processes [33]. Specifically, “the structural patterns, or typology, of the network can have significant impacts on how actors behave” [33] and how actors ultimately make decisions with respect to resource management. SNA involves the mapping (sociograph) and empirical assessment of communication pathways and interconnections (i.e., how knowledge is created and exchanged) that exist and emerge between actors within a bounded social system. For the purposes of this study, a social network refers to the communication and relationships that exist, or develop, between actors participating in decision-making processes that are related to the sustainable management of common- pool resources.

A social network is comprised of a set of actors, whether individuals or aggregated groups, linked through one or more relationships [34] [35]. Actors are referred to as "vertices" or "nodes" in the network, and the relationships between actors are referred to as "edges" or "links". The latter are associated with communication mechanisms, or information exchange pathways. Communication between the actors defines both the network and the social network data. For each of the two case-study watersheds, a survey was conducted to identify the actors and their relationships, followed by a formal network analysis [35] [36]. An adjacency matrix was constructed for each of the watersheds, using binary data, to represent an existing link or relationship between any two actors (each pair of potentially linked actors is called a dyad). The relationships were then analyzed through the use of structural algorithms embedded in the NodeXL [36] SNA software.

This study, through use of the Adaptive Management (AM) – Network Structure Typology Framework [30], examined the often-implicit relationship between structure and functionality of the social networks in the two case study watersheds. The framework was employed to guide the collection of social network data through the specific metrics of reachability and closeness centrality and considered key quantitative metrics for measuring adaptive capacity within an AM network [30]. The betweenness metric was also incorporated to enable the identification of actors holding a key position within the network, which is considered the optimum for connecting disparate actors and communities (clusters) within the network.

As part of the sense-making SNA process, network actors were broadly defined as actors participating in the watershed planning process (e.g., expert advisors, interested watershed residents, technical committee members, advisory committees, water users (licensees), and identified government personnel, general stakeholders, researchers etc.). The SNA survey tool was employed to capture social network data from both the formal and informal actors (i.e., not identified in the formal list, but identified by formal actors as contributing helpful water governance related information) who were involved in the watershed planning processes in both the Similkameen and Kettle watersheds. The total number of actors, including both formal actors identified by regional district governments and informal actors identified through the survey, included 59 actors (n=59) for the Similkameen Valley Watershed (SVW) planning process and 54 (n=54) actors for the Kettle River Watershed (KRW).

All individuals identified by the regional governments as formally participating in the watershed planning process were included in the initial bounding list for the process. Formal network actors, those identified by the Regional District as members involved in the watershed planning process through committee involvement, were asked to identify up to 10 other members (formal) with whom they interacted the most and whose interaction was the most important as far as assisting in their watershed governance decision-making. The formal network actors were also requested to identify up to five additional people who were not a part of the formal network (informal network), but who influenced their water governance decision-making through knowledge provision. Through this process, both the formal and informal (egocentric) networks for each of the participants was captured. These limits, ten formal and five informal, were determined through a review of existing SNA surveys, which determined these quotas as a good balance between strong to weak ties versus ease of completion of survey for respondents.

Watershed actors remained anonymous and were identified by the organizations, or level of government, they represented, with the exception of actors who self-identified as First Nations or representing a First Nations Band. The total number of responses (n), including both online survey responses and interview responses reached 27 out of a total 33 (82%) for the Similkameen and 25 respondents out of a total network (formal and informal) size of 36 (70%) for the Kettle.

6.2.3 Graphlet Analysis

In this analysis, correlations between graphlets (detailed below) were defined and utilized to create a new and richer network measure that, unlike other measures, makes network structure directly interpretable and provides clear translation into everyday language. As such, this new measure can uncover novel relationships between seemingly unrelated networks from different domains. Furthermore, it can be used to track the dynamics and explain the evolution of any network.

For comparison with social network analysis (SNA) methods, two graphlet techniques were employed: graphlet degree vector (GDV) and graphlet correlation matrix (GCM) [20]. A graphlet degree vector for a given node measures the number of times that a given node is touched by a particular orbit of a graphlet. For undirected graphs, a graphlet degree vector represents all automorphism orbits for 2-5 node graphlets, resulting in 73 entries. It has been shown that redundancies can be removed in graphlet degree vectors by exploiting dependencies between orbit counts in a network [4]. This results in a decrease of the orbit count, from 73 to 56, for 2-5 node graphlets. If the 5-node graphlets are excluded, this non-redundant orbit count is further reduced to 11. There are fewer dependencies between the 11 non-redundant 4-node orbits than the 56 non-redundant 5- node orbits. The 4-node orbits introduce less statistical noise, hence the use of a 2-4 node GDV was chosen.

This same concept was further extended to directed graphlets in [5]. Due to the nature of directed graphlets, there are many more combinations of graphlets, depending on how the edges are oriented. It was further revealed that, just as in undirected graphlets, directed graphlets provide useful results, while considering graphlets only up to a size 3. Since the nature of the network in this case is directed, directed graphlet metrics were used to generate the Graphlet Degree Vectors. After elimination of the redundant orbits, 10 non-redundant 3-node directed graphlet orbits remain.

Using the GDVs for each node, a matrix was constructed in which each row represents a vector. This matrix is used to construct a GCM by calculating Spearman's Rank Correlation

Coefficient between each possible pair of columns, which results in a 10x10 matrix that summarizes the topology of the network. Different real and model networks generally have very different orbit dependencies and hence very different GCMs [20].

Taking this another step further, two GCMs can be compared by calculating the Euclidean distance between the upper triangle of the two matrices. This results in another matrix term, the graphlet correlation distance (GCD), and in this particular case, DGCD-10. A directed network of any size can have its topology encoded into a 10x10 GCM and be compared with another network, or the same network over time, using a DGCD-10, thus making comparisons very computationally efficient.

For every node, various types of centralities were calculated using social network analysis including betweenness, closeness, and eigen- vector. In addition to centrality measures, cluster analysis was also performed using the Girvan and Newman (2002) clustering algorithm, which produced a clustering coefficient for each node. The centrality measures helped to facilitate identification of bridging actors, while the cluster analysis provided additional context to the node's role in the network. Using the same dataset, the GDV for every node in the matrix was calculated. Spearman's Rank was used to compare the betweenness values to the GDV values.

Every node was given a centrality rank, which represented the centrality score relative to the other nodes in the network. The node with the highest centrality score was given rank one, the next highest was given rank two, and so on. Similarly, each node was given a rank for each orbit in its graphlet degree vector. For orbit 0, the node with the highest orbit participation score was given rank 1, and so on. This was calculated for all 10 directed orbits. The same centrality rank of the nodes was compared to each orbit ranking separately. The same analysis was also performed for the clustering coefficient and orbit counts. Both watershed governance networks show positive correlations between orbit rank and the various centrality ranks.

6.2.4 Watershed Governance Network Analysis

Though sophisticated methods for analyzing complex networks have the potential to be of great benefit to almost all scientific disciplines, further work is required. In this work, we make fundamental methodological advances towards this. We discovered that the interaction between a small number of roles, played by nodes in a network, can characterize a network's structure and can also provide a clear, real-world interpretation. Given this insight, a framework for analyzing and comparing networks was developed, which has not yet existed for counting directed graphlets. Its strength has been demonstrated by uncovering novel relationships between seemingly unrelated networks, such as Facebook, metabolic, and protein structure networks and through its use to track the dynamics of the world trade network, which found that a country's role of a broker between non-trading countries indicates economic prosperity, whereas peripheral roles are associated with poverty. This result, though intuitive, has escaped all existing frameworks. Finally, the approach translates network topology into everyday language, bringing network analysis closer to domain scientists.

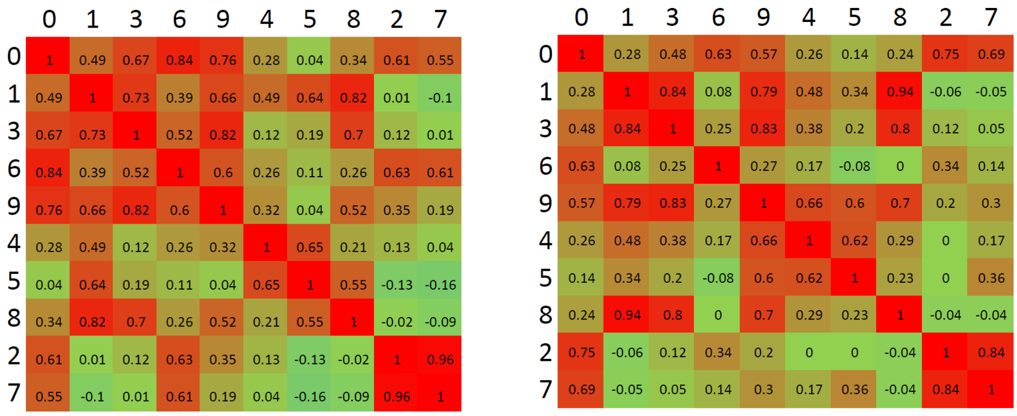
Graphlets have shown promise when applied to social networks. GCD-11 has been applied to the Facebook social network of several universities and been compared with several other networks, both real and synthetic. It was shown that, with Facebook networks, as well as metabolic and protein structure networks, they are best fit by geometric random graphs (GEO), geometric graphs that mimic gene duplications and mutations (GEO-GD), and scale-free networks that also mimic gene duplications and mutations (SF-GD) [20]. This result is surprising and offers insight into how Facebook networks grow over time.

The world trade network (WTN) is another real world network that has been compared using GCD-11. Specific orbits within the 11 non-redundant orbits that make up the GDVs and GCMs are more connected than others. Some orbits lie on the periphery, while others of higher degree are more clustered. GCMs of the trade network revealed that these two clusters are not correlated at all for this network. From this observation, it can be concluded that countries can either be a broker, meaning the node representing the country is touched by more clustered orbits, or on the periphery, but not both. This contrasts with biological networks, where nodes can participate in both clustered and peripheral orbits. A further investigation was done to determine if the wiring, or topology, of the WTN changes with variation in the price of crude oil. Through a comparison of the WTN, using GCD-11, and oil prices over the time period of 1962 to 2010, it was found that a strong correlation is present with changes in network topology and variation in oil prices. [5] also compared the results of directed graphlet counting with undirected graphlet on the WTN and revealed that directed graphlets provide a significantly richer and noise-free result.

Since graphlets are a new technology, there are no widely available tools to perform graphlet analysis on a network. The tools that do exist are concealed behind robust and complicated statistical software solutions written in R with custom libraries. A strong understanding of R is required to begin analysis, and even further knowledge is required to interpret results and modify the results to a usable format. Because of this, an application to perform all required calculations to produce directed graphlet and orbit counts was developed for this project.

All four required steps are completely separate to produce a final result of a DGCD-10. Networks are input in Pajek format as a text or CSV file. At any step, the data can be extracted in an easy to use CSV format. In the case of GCMs and DGCD-10s, the data can be output as a heatmap matrix or a matrix of values representing Spearman's Rank or Euclidean distance respectively (Figure 6.4(a) and Figure 6.4(b)).

Graphlets do not suffer from these drawbacks and can be used to define node roles and to design methods for linking the network structure with real-world function. They are defined as small induced subgraphs of a large network that appear at any frequency and hence are independent of a null model (denoted by directed graphlets G0 to G5 in Figure 1.1). An induced subgraph means that, once the nodes are chosen in the large network, all the edges between them must be chosen to form the subgraph. The correlations between graphlets are defined and utilized to create a superior network measure that, unlike other



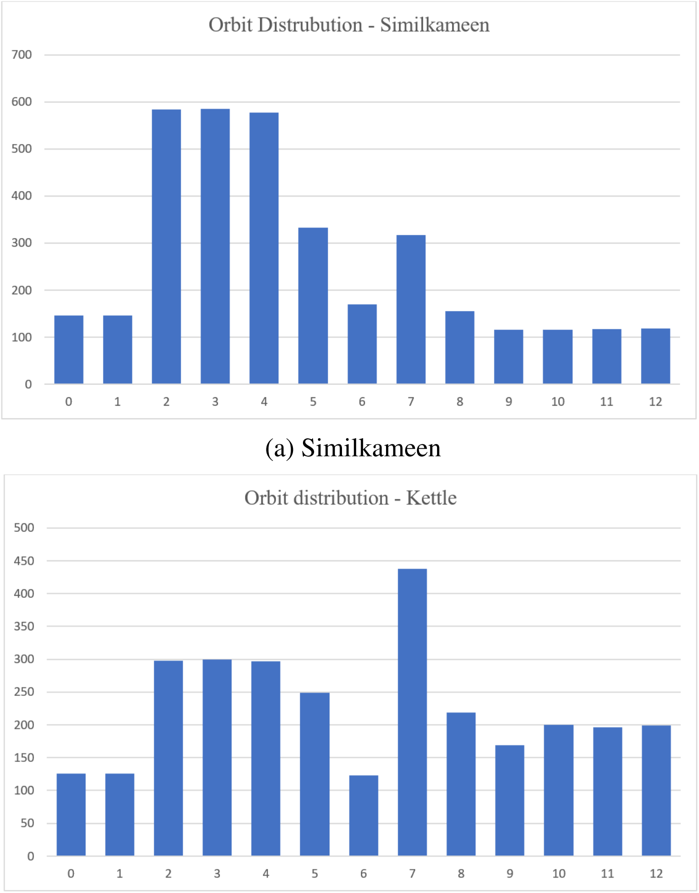
	0	1	3	6	9	4	5	8	2	7
0	1	0.49	0.67	0.84	0.76	0.28	0.04	0.34	0.61	0.55
1	0.49	1	0.73	0.39	0.66	0.49	0.64	0.82	0.01	-0.1
3	0.67	0.73	1	0.52	0.82	0.12	0.19	0.7	0.12	0.01
6	0.84	0.39	0.52	1	0.6	0.26	0.11	0.26	0.63	0.61
9	0.76	0.66	0.82	0.6	1	0.32	0.04	0.52	0.35	0.19
4	0.28	0.49	0.12	0.26	0.32	1	0.65	0.21	0.13	0.04
5	0.04	0.64	0.19	0.11	0.04	0.65	1	0.55	-0.13	-0.16
8	0.34	0.82	0.7	0.26	0.52	0.21	0.55	1	-0.02	-0.09
2	0.61	0.01	0.12	0.63	0.35	0.13	-0.13	-0.02	1	0.96
7	0.55	-0.1	0.01	0.61	0.19	0.04	-0.16	-0.09	0.96	1

(a) Similkameen

	0	1	3	6	9	4	5	8	2	7
0	1	0.28	0.48	0.63	0.57	0.26	0.14	0.24	0.75	0.69
1	0.28	1	0.84	0.08	0.79	0.48	0.34	0.94	-0.06	-0.05
3	0.48	0.84	1	0.25	0.83	0.38	0.2	0.8	0.12	0.05
6	0.63	0.08	0.25	1	0.27	0.17	-0.08	0	0.34	0.14
9	0.57	0.79	0.83	0.27	1	0.66	0.6	0.7	0.2	0.3
4	0.26	0.48	0.38	0.17	0.66	1	0.62	0.29	0	0.17
5	0.14	0.34	0.2	-0.08	0.6	0.62	1	0.23	0	0.36
8	0.24	0.94	0.8	0	0.7	0.29	0.23	1	-0.04	-0.04
2	0.75	-0.06	0.12	0.34	0.2	0	0	-0.04	1	0.84
7	0.69	-0.05	0.05	0.14	0.3	0.17	0.36	-0.04	0.84	1

(b) Kettle

Figure 6.4: GCM Serriated + Redundancy removed



(a) Similkameen

(b) Kettle

Figure 6.5: Cumulative orbit counts

simple or complex measures, makes network structure directly interpretable and provides its clear translation into everyday language. As such, this new measure can uncover novel relationships between seemingly unrelated networks from different domains. Furthermore, it can be used to track the dynamics and explain the evolution of any network, which, as demonstrated with the World Trade Network example. We computed the modularity indices for the two networks and provide them in Table 6.3.

Table 6.3: Modularity Index for Figure 6.3(a,b)

	Similkameen Network	**Kettle River Network**
Modularity Index	0.365	0.277

6.2.5 Results and Observations

Water governance often adopts one of two end-member frameworks: (a) centralized, command & control structures, or (b) distributed collaborative networks. The former typifies the traditional style of water governance that has reigned for the past century, whereas the latter is increasingly touted as a panacea to the evolving challenges of water resource management in a time of rapidly changing drivers (e.g., climate change, urbanization). This study applies Social Network Analysis (SNA) to two case-study watersheds in south central British Columbia in order to assess the (mis)alignment between water governance network structure and stakeholder objectives regarding adaptation to the pressures imposed by climate change.

The results indicate that rural, water-scarce regions continue to be burdened by centralized, command- control style structures that reinforce the status quo in watershed governance [37]. This reality marginalizes stakeholders at the peripheries of the network, who may represent a silent, but significant, voice in regard to future visions for watershed governance. The management of common-pool resources in rural areas will likely remain a difficult challenge without social networks that are designed strategically so as to become better aligned with stakeholder visions.

Since our objective is to compare the SNA measures with graphlets measures, we used the same technique to determine the correlation between the results in both analysis. That is, Spearman's Rank Correlation Coefficient is used to determine the correlation between the results in both analysis.

The comparison only uses a subset of the tools that graphlet analysis provides, namely that of the graphlet degrees for each node, and omits graphlet correlation matrices and graphlet correlation distance. The strong correlation between the SNA measures and the graphlet measures chosen is a strong motivator for further analysis of the full graphlet toolset. Access to the SNA measures for betweenness, etc. were provided for each node in both the Kettle and Similkameen data sets. The program used enabled the calculation of the directed graphlet degree vector for every node, which provided a graphlet degree for each node. The graphlet degree vector, comprised of graphlet degrees for various orbits, was, in this case comprised of a set of 13 data points for each node. This value was compared with the value for the various SNA measurements and compared with both eigenvector and betweenness centrality measurements.

Once the values were calculated, the comparison was done first by ranking each node based on its score for both the SNA measure and the graphlet degree. This process was repeated for each orbit in the GDV. With a ranking for each SNA measure and a ranking for the each graphlet degree per node, the correlation coefficient was calculated. Further, the average of the ranks for all nodes was calculated in order to determine a more general comparison over the granularity that the node-by-node comparison provides. Figure 6.6 shows those average ranks for SNA measures as compared with graphlet analysis.

Observations resulting from directed graphlet analysis include the following:

- *Observation 1:* Comparing Figure 6.5(a) and Figure 6.5(b), we can see that the distribution for orbits 6,8 and 5,7 suggests importance of divergent versus convergent nodes. There are twice as many converging graphlet structures (G3) than diverging graphlet structures (G2) in the Kettle network. Whereas for Similkameen, the distribution of these structures are even.

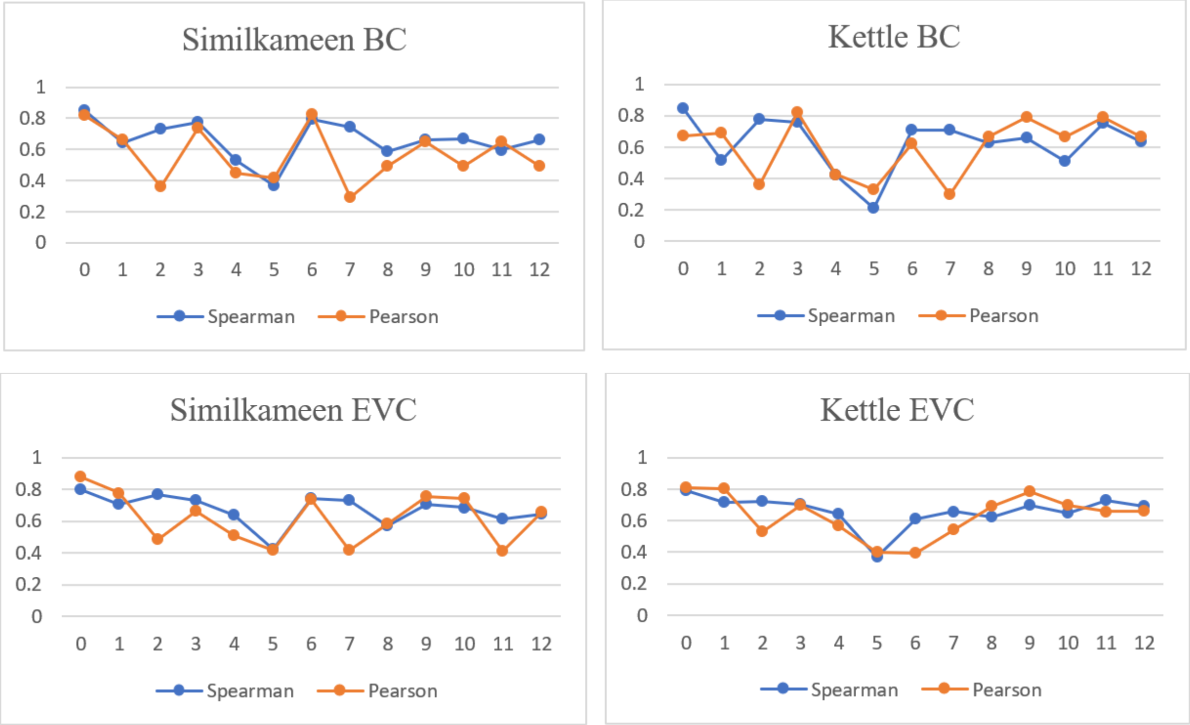


Figure 6.6: Correlation (Orbit counts vs Betweenness Centrality and Eigenvector Centrality)

Research has shown that within these rural watershed planning processes, where climate-change impact has been identified as a key challenge to water sustainability:

1. centralized, command and control (CC)-style topologies persist, contrary to the inclusive, collaborative models that are recommended to address climate-change impacts;
2. the core-periphery network structure that was evident in the two case-study watersheds is associated with classic challenges of communication and information-exchange, due to core actors remaining strongly linked to each other and weakly linked to periphery actors;
3. the role of bridging actors and organizations (BAOs), while critical in adaptive systems, is under-utilized or ineffectual in addressing the identified fragmentation challenge; and

4. due to periphery isolation (particularly amongst actors self-identifying as First Nations), network access to alternative knowledge bases and to innovative ideas and solutions remains limited.

In combination, these challenges suggest that successful implementation of newly developed water policy recommendations may be hampered due to isolated actors (e.g., First Nations, senior government) and "active non-participating" key industry actors (e.g., agricultural representatives) having little to no representation or voice in the process. Overall, the research demonstrates that, while water governance theory has evolved rapidly over the past decade, in practice, institutional inertia continues to favour the status quo in B.C., hindering any transition to inclusive (e.g., collaborative, polycentric, multi-scale, distributed, adaptive) water governance.

- *Observation 2:* GCM gives us a scale free representation of any network with structural properties encoded within it and its heatmap visualizes the same. The biggest advantage of having such a heatmap is for network comparisons on the bases of their structure. Network models such as hub and spoke or geometric networks, exhibit unique graphlet characteristics. Having their GCM gives us a scale of how close other networks behave to the existing network models that we are aware of.

  This is a very powerful comparison because this would reveal which of the two watershed governance networks are structurally closer to the unique models we are aware of. We can see in the heatmaps Figure 6.4(a,b) that there are both similarities and differences in the network. Assuming we have a governance model in future which we know work's well due to the organizational structure, we can then quantify which of these are structurally closer to the optimal model. Furthermore, we can also contrast these comparisons with other metrics such as Modularity Index (Table 6.3) and see how they correlate.

- *Observation 3:* In the contemporary context, water governance involves a growing diversity of actors and voices engaged in policy and plan development Although this is a positive shift in the global approach to resources management – one that allows alternative perspectives and ideas to enter the dialogue – there continues to be associated challenges with respect to communication barriers and information exchange, as the environment in which we manage water is becoming increasing complex. Social networks play important roles in connecting and organizing actors as they strive to develop collaborative strategies to attain natural resource goals particularly during times of uncertainty and rapid change [30] [38].

It has been argued that the distributed, also referred to as mesh or collaborative, network typology is preferable for adaptive governance due to more effective communication along multiple edges and increased levels of trust, thereby facilitating greater access to a wide variety of knowledge within the network [39]. Distributed network typologies such as appear better suited to address complex tasks due to the increased level of innovation resulting from a diversity of inter-connected actors [33].

It was observed that in the Similkameen network, communication is more like a flow (Graphlet G1), whereas in the Kettle network, communication appears more oriented to a central hub structure (G2, G3). In the Kettle, there were more than twice as many nodes receiving information than transmitting limiting the free flow of information amongst many actors within the network. In contrast to the Similkameen, the Kettle's core-periphery network typology is characterized by centralized decision-making and restricted communication pathways, leading to limited knowledge diversity and homogeneous values [33].

- *Observation 4:* Correlation between orbit counts and betweenness centrality is "generally" going to be positive. This is because since centrality measures the total number of shortest paths that pass through a vertex, higher centrality implies higher participation in graphlets. This however will not always be the case. For example, having two large cliques, connected by a narrow path implied that nodes on that path would have high betweenness centrality but low orbit participation. This would result in negative correlation between centrality and orbit count metrics. Since that is not the case for our watershed governance networks we can conclude that there are low narrow pathways for communication. As a result of more communication avenues for passing of information, points of failure of communication is low for these networks.

- *Observation 5:* There are three times as many transitive 3-cycles (G5) than intransitive cycles (G4). This is an interesting observation. However, what the implication of this would be is still not clear. Further inspection in future might be able to show that certain communication networks orient themselves to be transitive while others could be the other way around. However, this is still an interesting observation that could have some significant implications.

6.3 Gene Expression Analysis

We are living in the era of unprecedented technological advancements in both hardware and software that compete and feed each other. Despite such amazing levels of advancements in technology and computational techniques, current data analysis techniques lack the ability to derive accurate and useful information from gene expression data. Since the introduction of microarray technology around the mid-1990s, collecting and analyzing gene expression (GE) data to identify bio-markers for clinical applications such as diagnosis, prognosis, and/or prediction of suitable treatment has been widely practiced.

In addition, the technology for collecting GE data is getting cheaper, faster, and easier to use, which has led to rapid increase of GE data availability. This trend demands effective techniques to model and analyze such vast amount of bioinformatic data. One of the popular techniques is to model the data as graphs in order to effectively represent important elements in the data and their interactions to reveal hidden links and patterns. To understand hidden patterns and compare different networks, motifs are typically enumerated and analyzed.

6.3.1 Contributions

This section has three main contributions:

1. A new and relatively simple computational technique based on signed graphlets to reduce GE data of any large size to a simple heatmap is proposed. The heatmaps obtained from GE data collected under various conditions, we believe, could be used as or used to derive biomarkers of cellular states represented by the data. A database of such biomarker signatures could increase the accuracy and efficacy of diagnosis, prognosis, and hence choosing suitable treatment options.

2. We demonstrate our approach by analyzing GE data obtained from three types of cell samples – one from healthy cells and two from two types of cancer cells (gastric cardia adenocarcinoma (GCA) and gastric non-cardia adenocarcinoma (GNCA)). The initial results are encouraging. We observe that, within each sample type, the heatmaps are noticeably invariant.

3. As a test of robustness of the proposed graphlet based approach to GE data analysis, we repeated the graphlet metrics computation with 6 sets of random samples from each type and constructed the heatmaps.

A unique feature of our approach is that it reduces a given GE data of any size to a constant size heatmap.

6.3.2 Related Works

WGCNA (Weighted Gene Co-expression Network Analysis) [40] has been widely used to study GE networks. Both graphlet based analysis proposed in this and WGCNA begin with the same type of GE network, but they differ significantly in the steps that follow and the target analysis aimed for. Figure 6.7 describes the steps involved in WGCNA.

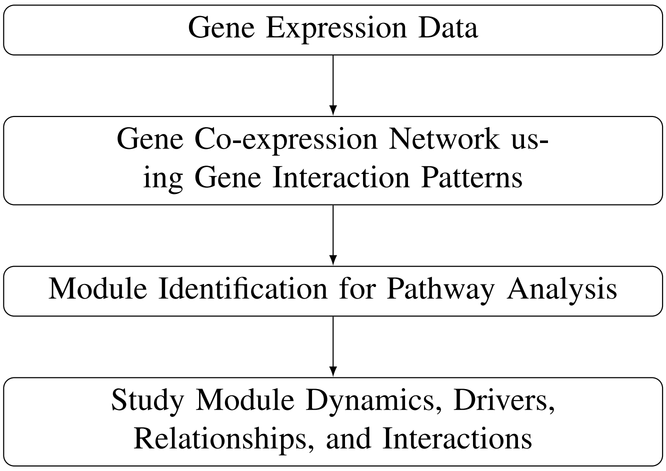

Gene Expression Data
Gene Co-expression Network using Gene Interaction Patterns
Module Identification for Pathway Analysis
Study Module Dynamics, Drivers, Relationships, and Interactions

Figure 6.7: WGCNA Approach

The proposed graphlet based analysis in comparison aims to capture the signature of the GE network more succinctly. In the case of signed graphlets, we are able to provide a fixed size signature of a network independent of a network's size and use that for further analysis and comparisons.

In the original work comparing gene expressions of gastric cancers given in [41], the results from the microarray experiments were analyzed using principal component analysis (PCA) and then paired t-tests were used to identify genes that were differentially expressed between the tumor samples and the matched healthy samples [41]. Genes that were considered to be differentially expressed had to have a 2-fold differential expression between tumor samples and healthy tissue and have a P-value less than 4.73×10^{-7} in the paired t-test. Based on the results obtained from [41], 511 dysregulated genes were identified.

6.3.3 Graphlet Metrics Computation

We begin signed graphlet analysis with a $N \times N$ correlation matrix. Using the algorithm presented in Section 3.2, we generate GOVs for each node. This information is an $N \times 15$ matrix. We go even further with this intermediate result and calculate correlation between the counts of the 15 signed graphlet orbits. This results in a 15×15 GCM. At this stage, we have effectively derived a fixes size matrix that captures valuable topological properties of a network irrespective of the size of the network. Since graphlet counts are not independent, we further identify 3 orbits in the case of signed graphlets for which the counts are redundant and finally result in a 12×12 GCM. We can then use these fixed size GCMs to compare networks and identify topological similarities and differences. Summarizing the above, we generate the graphlet metrics in the following order:

- Signed Graph $(N \times N)$
- Graphlets and Orbit counts $(N \times 15)$
- GCM and Heatmap (12×12)
- Graphlet Correlation Distance

6.3.4 Proposed GE Data Analysis Technique

The proposed approach to analyze GE data involves five major steps: computing Gene Co-expression Matrix, converting to signed network, counting signed graphlets and orbits, and generating signed graphlet metrics as shown in Figure 6.8.

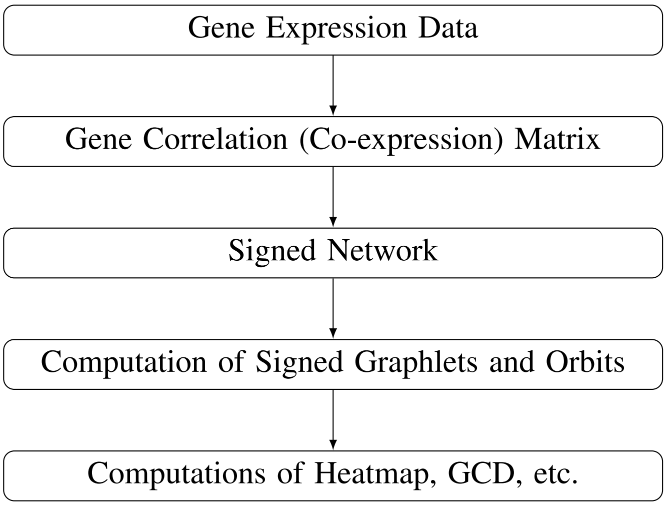

Gene Expression Data
Gene Correlation (Co-expression) Matrix
Signed Network
Computation of Signed Graphlets and Orbits
Computations of Heatmap, GCD, etc.

Figure 6.8: Signed Graphlet Based GE Analysis

The computations involved in the steps shown in Figure 6.8 are explained as follows.

1. *Gene Expression Data:* Gene expression data is typically available through online databases and repositories most of which is for public access. Gene expression data is a two dimensional array with rows as samples and columns as gene probe labels. Each cell (i,j) has the expression level of probe j in the sample i.

2. *Gene Correlation Matrix:* Since the Gene Expression Data from microarrays can have samples that can fall into multiple categories such as healthy, cancerous, infected, etc., we have to group the samples based on each category. Once Grouped, we take the Spearman's coefficient for each pair of gene probes and see how they correlate. This gives us a $N \times N$ correlation matrix where N is the number of genes in the data and each cell have a value in the range $[-1,1]$.

3. *Signed Network:* Depending on the number of samples and variance of the data, the GCM will be non zero, that is, there would exist some correlation between the expression values of the gene probes. We then proceed to convert this into a signed network by choosing suitable thresholds for positive and negative edges.

$$adj_{i,j} = \begin{cases} 1 & corr_{i,j} \geq \tau \\ -1 & corr_{i,j} \leq -\tau \\ 0 & otherwise \end{cases}$$

4. *Signed Graphlets and Orbit:* The process of counting signed graphlets and orbits are described in Chapter 2. We follow the steps which results in a Graphlet Orbit Vector for each node in the network. The length of this vector is the same as the total number of orbits for a type of graphlet. So in the case of 3 node signed graphlets, the length of the vector would be 15.

5. *Final GCM, Heatmap, GCD:* Given the orbit counts for the data, we then proceed to

represent how theses counts themselves correlate with one another as a GCM. This step is described in greater detail in Chapter 2. Finally, we end up with a 12×12 GCM.

Since the values in GCM is in the range $[-1, 1]$ we can easily express this as a heatmap to make it easier to observe patterns and differences. We also quantify the structural difference between two networks using the Graphlet Correlation Distance metric which as mentioned previously, takes the Euclidean distance of the upper triangles of a pair of GCMs.

6.3.5 Graphlet based Cancer GE Data Analysis

Here we describe the background in which the data is collected, filtering, and finally analyzed using our proposed graphlet based approach.

A: Data Collection to Signed Network Construction

Data collected starts with cell sample extraction from patients and then using microarray technology to obtain raw microarray data and finally obtaining GE data as explained next.

- *Cell Sample Background:* To obtain GE microarray data, cancer and healthy cell samples were taken mostly from male patients whose ancestry was from Shanxi province in China. The samples were presented to the Shanxi Cancer Hospital between 1998-2001 with a diagnosis of GCA and GNCA. Tumor and healthy samples from these patients were surgically obtained and then frozen in liquid nitrogen and stored at -130C. In total, 62 samples of GCA and 72 samples of GNCA and their healthy matched 134 samples were analyzed [41].

- *Cancer Characteristics:* Most of the samples were from late stage, high grade, and intestinal cell type tumors with lymph node metastasis. The average age of diagnosis was mid-to-late 50s. A quarter of samples taken were from patients who had first, second, or third degree relatives who also had upper gastrointestinal cancer.

- *Cell Sample to Microarray Data:* Samples to be used for microarray analysis had to fulfill the criteria of: (1) histological diagnosis of GCA or GNCA; (2) the samples to be used as tumor samples had to be composed of at least 50% cancerous cells; and (3) the RNA quality of the samples had to be adequate for testing. RNA was then extracted and purified from the tissue samples, after which RNA quality and quantity were determined. The microarray experiments were conducted using 8ug total RNA according to the Affymetrix microarray protocol. The microarray gene expression data can be obtained from https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-1307/.

- *Raw Microarray Data to GE Data:* Obtaining GE data from raw microarray data is a complex process. The original data consists of the 22,283 gene probe sets on the Affymetrix GeneChip for healthy and cancerous gastric cell samples. From 22,283 gene probes, the authors of [41] identified 511 dysregulated genes that corresponds to 612 gene probes. Microarray data is represented as an ARFF file.

- *Data Selection and Filtering:* The microarray gene expression data is first preprocessed probe set focussed expression levels instead of sample focussed. This processed data is then used to compute a Correlation Matrix. We have represented some of the signed networks using a force directed model in Figure 6.9 (bottom).

B: Graphlet Metrics Computation

In our experiment we start by considering N = 22,283 for all gene probe sets and N = 612 for the significant gene probe sets. Since the correlation values are in the range [-1,1], we chose a threshold which would appropriately lead to a non complete or non empty network. As per Subsection 6.3.4, we chose a threshold of $(> +0.6)$ and (< -0.6) for positive and negative edges on the bases of network structure of the healthy cell and followed the same threshold for the other cell categories.

With the given signed network, we then proceed to compute graphlet and orbit counts.

This is similar to the steps carried out in Chapter 2. We do this for all 22,283 probes as well as for the identified subset of 612 gene probes for all samples. In addition we also compute graphlet and orbit metrics for 612 gene probes with 6 sets of 40 random samples for each category (Healthy, GCA, GNCA). At this stage, we are able to capture orbit participation for each gene probe set in our network. These GOVs can also be used to study genes which are of interest. The next step is to take our resulting GOV and coalesce it in the form of the graphlet correlation matrix. These are represented as a heatmap as shown in Figure 6.9 (top). We do not consider all orbits since some their counts are related. To fix this, we identify the redundant orbits (10,11 and 14 in this case) and exclude them from our results.

C: Study Objective

Our main objective in this work is to derive biomarkers that could be used to identify and distinguish different cell states. Gene probe sets in different samples can express in varying degrees depending on many factors that affect the state of the cell. There can be minor factors such as time of day, conditions of testing, etc. However, the major factors such as diseased or healthy that we want to be able to capture and highlight. In order to test the sensitivity of the result and to ensure that minor factors do not skew it, we have performed random sampling of our data. Choosing 40 samples at random from all the samples. In summary, we want to observe the variation (and invariance) between the signatures of:

1. all 22,283 gene probes and 612 gene probes, for all samples and all cell types.
2. 612 gene probes for random subsets of samples and all cell types.

To achieve our objective, we computed GCM and corresponding heatmaps using:

- 22,283 genes probes for 134 healthy cell samples, 62 GCA cell samples, and 72 GNCA cell samples.
- 612 gene probes 134 healthy cell samples, 62 GCA cell samples, and 72 GNCA cell samples.

- 612 gene probes for random 40 healthy cell samples, random 40 GCA cell samples, and random 40 GNCA cell samples. (This computation is done for 6 random sets of 40 samples each.)

In total, we computed 30 heatmaps, 3 signed networks of 612 gene probes as nodes, and the mean correlation distance between the GCMs of the random samples for all three types of cells. The graphical results are shown Figure 6.9 and Figure 6.10.

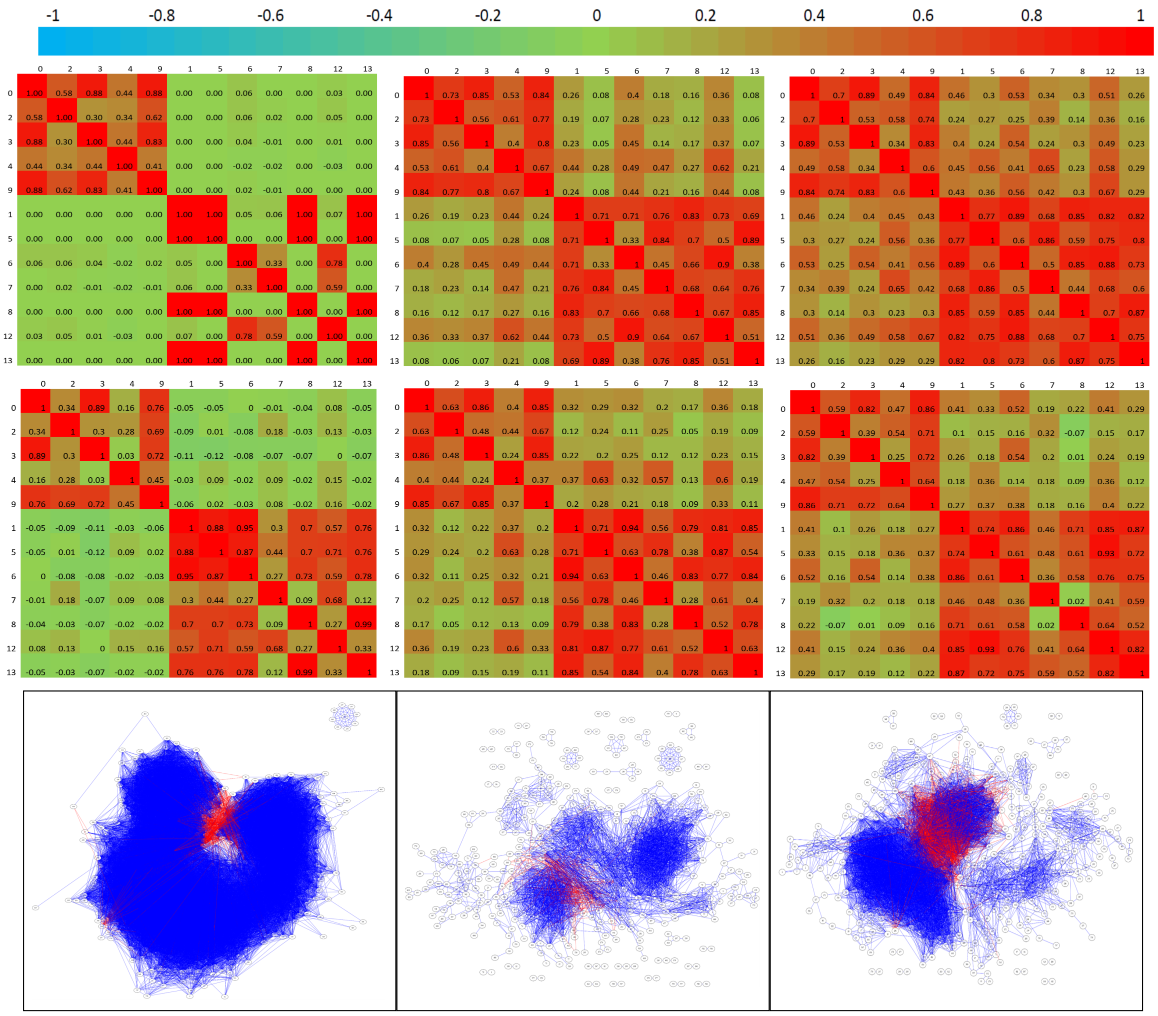


Figure 6.9: H_611_60, GCA_611_60, GNCA_611_60 (All samples, left to right; top - heatmaps, bottom - network)

D: Observations

From the experiments we are able to observe:

1. A clear visual distinction in terms of signed networks between three types of cells – healthy, GCA, and GCNA. Also, the intensity of correlation between gene probe sets in a healthy cell is much higher than the cancerous cell. As a result, we can see that for the same threshold, the densities of the network in all three categories vary. This observation might be a representation of differences in gene regulatory control between cancerous and healthy cells. In particular, we observe what appears to be less regulation in cancer cells as indicated by the sparse edges (co-regulation) in the cancerous cell network whereas the healthy cell has a dense network that could indicate high level of gene regulation. This is consistent with the general understanding of the differences in gene regulation between healthy cells and cancerous cells.

2. An invariant pattern between the heatmap generated using 22,283 gene probes and 612 gene probes of same type of cells. That is, the patterns observed in all gene probes is preserved in the patterns observed in the small subset of significant gene probes. So, the absence of other gene probes has not affected the overall pattern of the heatmap.

3. We see some general patterns in all GCMs. For instance, we can see clustering of orbits (0,2,3,4,9) and orbits (1,5,6,7,8,12,13). This however will not always be the case for all signed networks. In Chapter 2, we have generated random signed networks with uniform distribution of edge signs and have observed different patterns. A possible reason behind this similarity could be that the cancerous cells are derived from gastric cells. As a result, some key structural similarity might be preserved that can be observed from the heatmap.

4. For a quantitative analysis, we computed mean GCD between networks of sampled data as shown in Table 6.4.

Table 6.4: Mean Graphlet Correlation Distance

	H	**GCA**	**GNCA**	**R_H**	**R_GCA**	**R_GNCA**
H	0	1.937	2.072	**0.789**	2.17	2.51
GCA	1.937	0	1.139	1.82	**0.781**	1.48
GNCA	2.072	1.139	0	1.91	1.01	**0.781**

From the table, we can see that the GCD average between the original and respective random samples of same cell type are most similar (low average values). The distance between healthy and cancerous cells are higher compared to the distance between the two types of cancers. This also can be seen visually from the heatmaps given in Figure 6.10. We can visually see that dominant patterns in the heatmap for each type of cell are almost intact.

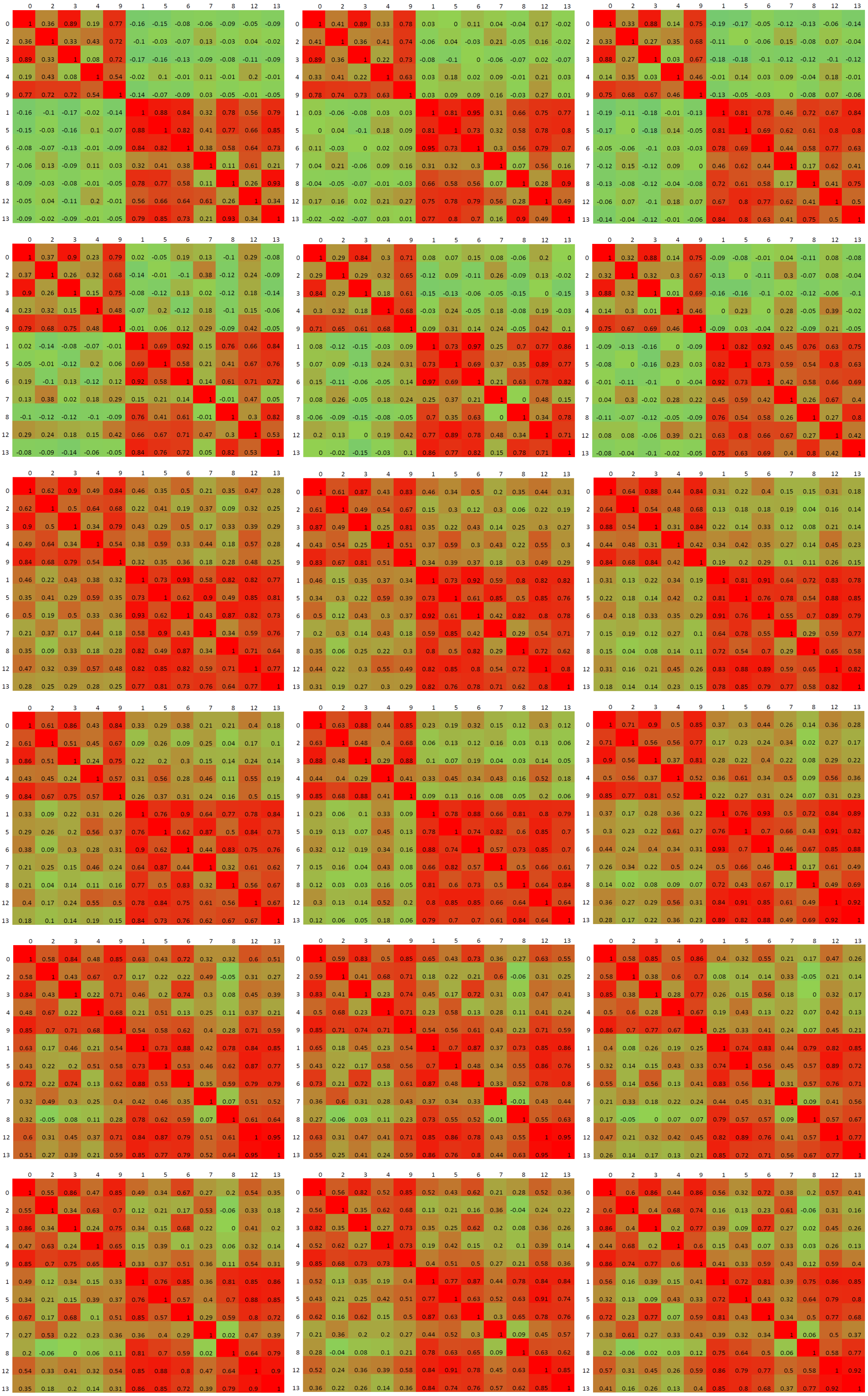


Figure 6.10: Random Sampling (40 Samples): R_H_611_60 (top 6); R_GCA_611_60 (middle 6); R_GNCA_611_60 (bottom 6)

Chapter 7

Summary and Future Work

In this thesis we first present an efficient generic algorithm for enumerating 3-node graphlets and orbits. It can count graphlets and orbits for undirected, directed, or even signed graphlets introduced recently in [6]. The algorithm is simple, and more importantly, to the best of our knowledge, is the first algorithm with its characteristics. The proposed algorithm, when compared with the existing best algorithms in its category, turns out to be very efficient. We also demonstrate the computational and application aspects of graphlet metrics.

To demonstrate the computational aspect of the metrics, we modeled and studied random networks. For an application, we used a metabolic network, derived from an enzyme and metabolite correlation network in corn.

Some new directions for improving on the algorithm and related work:

1. The algorithm can be extended for higher order graphlets. Since this algorithm is parallelized, it could be easily implemented in a distributed system environment for even better processing power for massive graph structures.

2. One of the improvements to this algorithm could be replacing the routine for counting triangles with the help of Strassen's algorithm (or the fastest matrix multiplication algorithm) and the adjacency matrix multiplication. However, this would require a $O(n^2)$ space complexity which is not feasible for big graphs, but may work well for

small dense graphs.

3. Another variation is to use an intersectionList just as we used differenceList. This would unify both the routines. However, experimentation has shown that this becomes a bottleneck in the case of dense graphs.

4. We can extend the work by applying the same analysis to other species, and with graphlets we can compare how similar or dissimilar the networks are among species.

Next, we study two watershed governance networks using directed graphlets. To the best of our knowledge, this is the first work that applied graphlets to explore watershed governance networks. The specific contributions here include:

1. Comparison of graphlet measures with centrality measures.

2. Graphlet orbit distribution and analysis of the two watershed networks.

3. Correlation matrices, heatmaps, and analysis of the two networks using them.

4. Network modularity index computation and analysis.

Here, we have used graphlets to both explore the structure of the individual network and compare different networks. We believe graphlet based network metrics can not only complement the social network metrics but can also provide new and richer results and insights. This can lead to deeper understanding of the current governance (watershed) and in turn provide improved directions for more effective changes to governance structures.

Finally, we present an application of signed graphlets in the domain of bioinformatics, specifically in studying patterns exhibited by the co-expression network of healthy and cancerous stomach cells. We found that this tool provided some interesting insights in co-expression networks and our testing in this field presents a proof of concept of its applicability.

Some new directions for related work:

1. A further analysis on the impact of graphlet frequencies as compared to orbit counts would help. It would allow us to understand as to when it would be more beneficial since counting orbit comes at an additional cost as compared to counting only graphlets.

2. We hope to extend this research further and test how these networks differ with changing sample conditions, temporal changes in biological processes and different types of cancer.

In addition we propose to work on a comprehensive web framework to count graphlets.

7.1 Graphlet Decomposition Framework

The following graphlet decomposition framework is aimed to be a fast C++ framework using previously existing algorithms for counting graphlets and/or orbits and to return the GDV, GFD, GCM, GCD and other useful metrics depending on user requirements. Based on the user data and requirements, some algorithms would be used over others to achive better performance. The algorithms would be parallelized where possible and this framework would provide a nice sandbox for experimentation of graphlet counting algorithms.[1]

Being a portable and fast library, the intent is to be able to deploy a framework using this library in the cloud such that users can feed their network information to the application and get back the graphlet and orbit count information for their network in return. Alternatively, since the project is open source, users can choose to build and run this directly on their system for graph analysis.

Other features such as graph visualization and GCM heatmap visualization can also be included in later iterations. Figure 7.1 provides a graphlet decomposition web-framework.

[1] This work is being pursued jointly with with *Mike Drakos*.

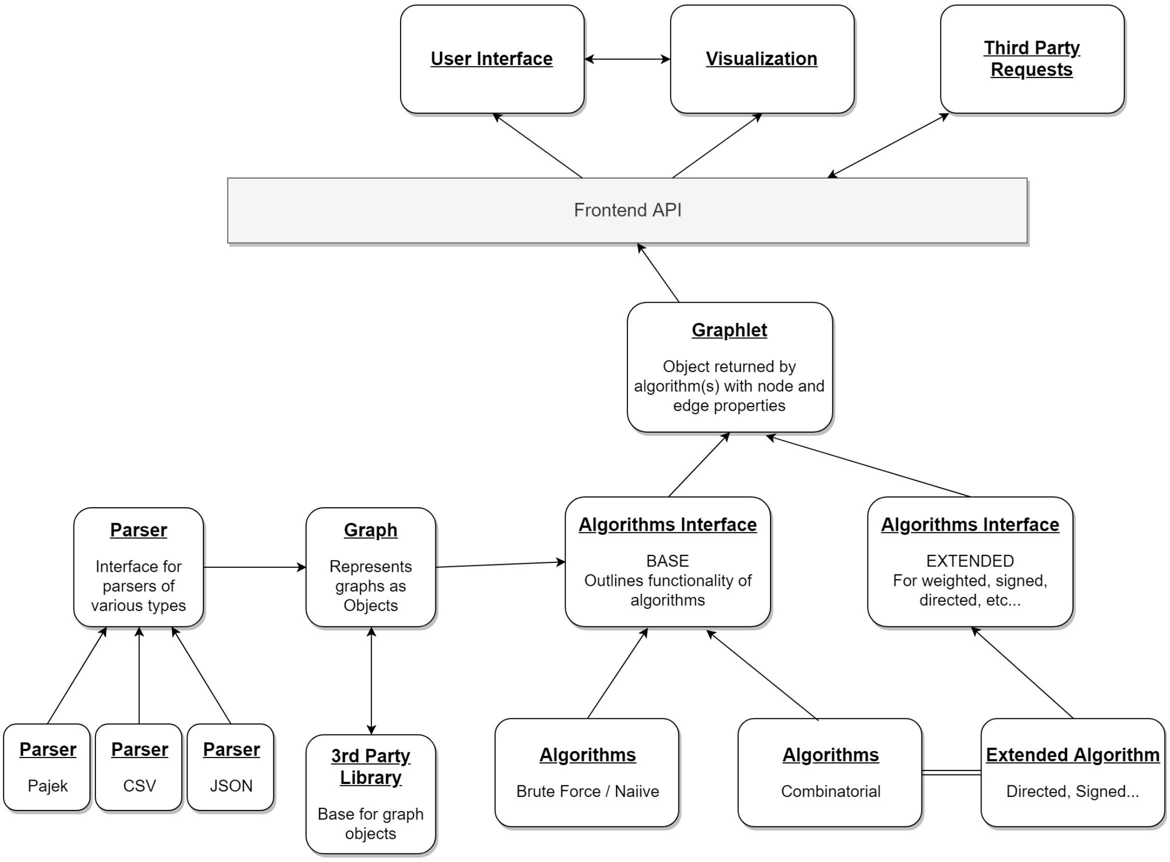


Figure 7.1: Graphlet Decomposition Web-Framework

Bibliography

[1] Siddharth Suri and Sergei Vassilvitskii. Counting triangles and the curse of the last reducer. In *Proceedings of the 20th international conference on World wide web*, pages 607–614. ACM, 2011.

[2] N. Pržulj, D. G. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004.

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

[4] Malod-Dognin N. Davis D. Levnajic Z. Janjic V. Karapandza R. Stojmirovic A. Yaveroğlu, Ö. N. and N. Pržulj. Revealing the hidden language of complex networks. *Nature Scientific Reports*, 4(4547):1–9, 2014.

[5] Anida Sarajlić, Noël Malod-Dognin, Ömer Nebil Yaveroğlu, and Nataša Pržulj. Graphlet-based characterization of directed networks. *Nature Scientific Reports*, 6:35098, 2016.

[6] M. Dale. *Applying Graph Theory in Ecological Research*, 2017.

[7] Alessandro Rinaldo Steffen Lauritzen and Kayvan Sadeghi. Random networks, graphical models andexchangeability. *J. R. Statist. Soc.B*, 80(3):481–508, 2018.

[8] Frank Harary and Edgar M. Palmer. Graphical enumeration. 1973.

[9] Darren Davis, Ömer Nebil Yaveroğlu, Noël Malod-Dognin, Aleksandar Stojmirovic, and Nataša Pržulj. Topology-function conservation in protein–protein interaction networks. *Bioinformatics*, 31(10):1632–1639, 2015.

[10] Arzu Burcak Sonmez and Tolga Can. Comparison of tissue/disease specific integrated networks using directed graphlet signatures. *BMC bioinformatics*, 18(4):135, 41–50, 2017.

[11] Nesreen K Ahmed, Jennifer Neville, Ryan A Rossi, Nick G Duffield, and Theodore L Willke. Graphlet decomposition: Framework, algorithms, and applications. *Knowledge and Information Systems*, 50(3):689–722, 2017.

[12] Tomaž Hočevar and Janez Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, 2014.

[13] Dror Marcus and Yuval Shavitt. Rage–a rapid graphlet enumerator for large networks. *Computer Networks*, 56(2):810–819, 2012.

[14] Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–24. ACM, 2008.

[15] Mihail N Kolountzakis, Gary L Miller, Richard Peng, and Charalampos E Tsourakakis. Efficient triangle counting in large graphs via degree-based vertex partitioning. *Internet Mathematics*, 8(1-2):161–185, 2012.

[16] Rasmus Pagh and Charalampos E Tsourakakis. Colorful triangle counting and a mapreduce implementation. *Information Processing Letters*, 112(7):277–281, 2012.

[17] Thomas Schank and Dorothea Wagner. Finding, counting and listing all triangles in large graphs, an experimental study. In *International workshop on experimental and efficient algorithms*, pages 606–609. Springer, 2005.

[18] Ryan A. Rossi and Nesreen K. Ahmed. Coloring large complex networks. pages 1–51, 2014.

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

[20] O. N. Yaveroglu. Graphlet correlations for network comparison and modelling: World trade network example. *PhD Thesis*, 2014.

[21] Tijana Milenković, Jason Lai, and Nataša Pržulj. Graphcrunch: a tool for large network analyses. *BMC bioinformatics*, 9(1):70, 2008.

[22] Tijana Milenković and Nataša Pržulj. Uncovering biological network function via graphlet degree signatures. *Cancer informatics*, 6:CIN–S680, 2008.

[23] Tijana Milenković, Vesna Memišević, Anand K Ganesan, and Nataša Pržulj. Systems-level cancer gene identification from protein interaction network topology applied to melanogenesis-related functional genomics data. *Journal of the Royal Society Interface*, 7(44):423–437, 2009.

[24] Paul W Holland and Samuel Leinhardt. Local structure in social networks. *Sociological methodology*, 7:1–45, 1976.

[25] Ömer Nebil Yaveroğlu, Noël Malod-Dognin, Darren Davis, Zoran Levnajic, Vuk Janjic, Rasa Karapandza, Aleksandar Stojmirovic, and Nataša Pržulj. Revealing the hidden language of complex networks. *Scientific reports*, 4:4547, 2014.

[26] Omer Nebil Yaveroglu. Graphlet correlations for network comparison and modelling: World trade network example. 2013.

[27] D. et. al. Toubiana. Correlation-based network analysis of metabolite and enzyme profiles reveals a role of citrate biosynthesis in modulating n and c metabolism in *zea mays*. *Frontiers in Plant Science*, 7(1022):1–10, 2016.

[28] W. Li and D. Schuurmans. Modular community detection in networks. *Proc. of the* 22^{nd} *Intl. Conf. on Artificial Intelligence*, pages 1366–1371, 2011.

[29] M. T. H. van Vliet and Wada Y. Florke, M. Quality matters for water scarcity. *Nature Geoscience*, 10:800–802, 2017.

[30] Crona B. Bodin, O. and H. Ernstson. Social networks in natural resource management: what is there to learn from a structural perspective? *Ecol. Soc.*, 11(2), 2006.

[31] Bauer B. Horning, D. and S. Cohen. Missing bridges: Social network (dis)connectivity in water governance. 2016.

[32] H. Hamilton. Similkameen river water management plan: Part 1-scoping study. 2011.

[33] O. Bodin and B.I. Crona. The role of social networks in natural resource governance: what relational patterns make a difference? *Glob. Environ. Change*, 19(3):366–374, 2009.

[34] Ernstson H. Stein, C. and J. Barron. A social network approach to analyzing water governance: the case of the mkindo catchment, tanzania. *Phys. Chem. Earth A/B/C*, 36(14e15):1085–1092, 2011.

[35] Shneiderman B. Milic-frayling N. Rodrigues E.M. Dunne C. Capone T. et al. Smith, M. Analyzing (social media) networks with nodexl. 2009.

[36] Shneiderman B. Hansen, D. and M. Smith. Analyzing social media networks with nodexl: Insights from a connected world. 2011.

[37] N. Andreas. Transforming rural water governance: Towards deliberative and polycentric models? *Water Alternatives*, 2(1):53–60, 2009.

[38] Elbe J. Elbe-S. Albrecht, M. and W. Meyer. Analyzing and evaluating regional governance networks: Three challenges for applications. evaluation. pages 20–58, 2014.

[39] K. J. Rathwell and G. D. Peterson. Connecting social networks with ecosystem services for watershed governance : A social-ecological network perspective highlights the critical role of bridging organizations. *Ecology and Society*, 17(2):24, 2012.

[40] Peter L. and Steve H. Wgcna: an r package for weighted correlation network analysis. *BMC Bioinformatics*, pages 9:559 doi:10.1186/1471–2105–9–559, 2008.

[41] Nan Hu Gangshi Wang and Howard H. Yang. Comparison of global gene expression of gastric cardia and noncardia cancers from a high-risk population in china. *PLOS One*, 8(5):1–10, 2013.