

The Betweenness Centrality Of Biological Networks

Shivaram Narayanan

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

T. M. Murali, Chair

Madhav Marathe

Anil Vullikanti

September 16, 2005

Blacksburg, Virginia

Keywords: Betweenness centrality, Vertex Betweenness, Edge Betweenness, Power
law, Biological networks

Copyright 2005, Shivaram Narayanan

A Study of Betweenness Centrality on Biological Networks

Shivaram Narayanan

(ABSTRACT)

In the last few years, large-scale experiments have generated genome-wide protein interaction networks for many organisms including *Saccharomyces cerevisiae* (baker's yeast), *Caenorhabditis elegans* (worm) and *Drosophila melanogaster* (fruit fly). In this thesis, we examine the vertex and edge betweenness centrality measures of these graphs. These measures capture how “central” a vertex or an edge is in the graph by considering the fraction of shortest paths that pass through that vertex or edge. Our primary observation is that the distribution of the vertex betweenness centrality follows a power law, but the distribution of the edge betweenness centrality has a “Poisson-like” distribution with a very sharp spike. To investigate this phenomenon, we generated random networks with degree distribution identical to those of the protein interaction networks. To our surprise, we found out that the random networks and the protein interaction networks had almost identical distribution of edge betweenness. We conjecture that the “Poisson-like” distribution of the edge betweenness centrality is the property of any graph whose degree distribution satisfies power law.

Acknowledgments

I would sincerely like to thank my advisor T. M. Murali, who has been one of the most patient and helpful guides one can ask for. Working on this thesis has been a great learning experience for me and I am grateful to him for giving me this opportunity.

I would like to thank Dr. Madhav Marathe and Dr. Anil Kumar Vullikanti for their valuable inputs, and numerous ideas that considerably improved this thesis.

I am also thankful to my parents, sister and family for all the love and support.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Contributions Of This Thesis	3
1.3	Readers Guide	4
2	Properties Of Biological Networks	5
2.1	Biological Experiments	6
2.1.1	The Two Hybrid System	6
2.1.2	Affinity Precipitation	7
2.1.3	Synthetic Lethality	8
2.2	Biological Datasets	8
2.2.1	BIND: Biomolecular Interaction Network Database	9

2.2.2	DIP: Database of Interacting Proteins	9
2.2.3	GRID: The General Repository for Interaction Datasets	9
2.3	Previous Results	10
3	Graph Generation Models	12
3.1	Erdős-Rényi Model	13
3.2	Barabási Albert Model	13
3.3	Watts Strogatz Small World Model	14
3.4	Eppstein Wang Model	15
4	Betweenness Centrality	17
4.1	Definition	17
4.2	Applications	19
4.2.1	Classification of scale free networks	19
4.2.2	Susceptibility of complex networks	20
4.2.3	Communities in Networks	21
5	Algorithms For Computing Betweenness Centrality	24
5.1	Background	24

5.2	Brandes Vertex Betweenness Algorithm	27
5.3	Newman Edge Betweenness Algorithm	29
6	Results	31
6.1	Vertex Betweenness	32
6.1.1	Vertex Betweenness Distribution	33
6.1.2	Vertex Betweenness vs. Degree Correlation	34
6.2	Edge Betweenness	37
6.2.1	Edge Betweenness Distribution	37
6.2.2	Edge Betweenness of Synthetically Lethal Interactions	41
6.3	Randomized Analysis	44
6.3.1	Simple Random Graphs	44
6.3.2	Eppstein Wang Power Law Random Graphs	46
6.4	Random Graphs with Similar Degree Distribution as the Biological Net- works	48
6.5	Further Analysis on Random Graphs	52
7	Conclusions	64

List of Figures

6.1	Vertex Betweenness Distribution for all three datasets	33
6.2	Vertex Betweenness values Vs Degree	35
6.3	Vertex Betweenness values Vs Degree	36
6.4	Mean and Standard Deviation of Vertex Betweenness values vs Degree	36
6.5	Edge Betweenness Distribution for Worm Network	38
6.6	Edge Betweenness Distribution for Yeast and Fly Networks	39
6.7	Edge Betweenness Distribution for all three networks	40
6.8	Edge Betweenness Distribution for Fly and Yeast Sub-Networks consid- ering the edge betweenness value from the original network.	42
6.9	Edge Betweenness Distribution for Fly and Yeast Sub-Networks	43
6.10	Edge Betweenness Distribution for the Simple Random Graphs	45

6.11 Edge Betweenness Distribution for the Eppstein Wang Power law Random Graphs	47
6.12 Edge Betweenness Distribution for 100 Random Graphs with Degree Distribution similar to Fly Network	49
6.13 Edge Betweenness Distribution for 100 Random Graphs with Degree Distribution similar to Yeast Network	50
6.14 Edge Betweenness Distribution for 100 Random Graphs with Degree Distribution similar to Worm Network	51
6.15 Average Edge Betweenness Distribution for Graphs with same size and density, for different values of power law exponent.	54
6.16 Average Edge Betweenness Distribution for Graphs with same size and density, for different values of power law exponent.	55
6.17 Average Edge Betweenness Distribution for Graphs with same size and density, for different values of power law exponent.	56
6.18 Average Edge Betweenness Distribution for Graphs with same size and density, for different values of power law exponent.	57
6.19 Average Edge Betweenness Distribution for Graphs with same size and density, for different values of power law exponent.	58

6.20 Average Edge Betweenness Distribution for Graphs with same size and density, for different values of power law exponent.	59
6.21 Average Edge Betweenness Distribution for Graphs with same size and power law exponent, for different values of density.	60
6.22 Average Edge Betweenness Distribution for Graphs with same size and power law exponent, for different values of density.	61
6.23 Power law exponent vs. Position of Spike	62
6.24 Density vs. Position of Spike	63

Chapter 1

Introduction

1.1 Motivation

Rapid advances in high-throughput and large scale experiments in biology are providing us with breathtaking new insights into cellular machinery and processes. The public availability of protein-protein interaction networks containing thousands of interactions for a number of species is a highlight of these advances. Studying the properties of protein interaction networks promises to yield predictions of protein functions,¹⁻⁵ shed new light on the evolution of biological networks⁶ and also improve our understanding of the structure of these networks.^{7,8}

It is natural to represent a protein interaction network as an undirected graph, where each vertex corresponds to a protein and each edge represents an interaction between

the two connected proteins. This representation allows us to examine the network from the point of view of graph theory. One of the first observed properties of protein interaction networks was that their degree distribution follows a power law,^{6,7} i.e., the fraction of vertices with degree k is proportional to $k^{-\gamma}$, for some $\gamma > 0$. A number of models have been proposed to explain how protein interaction networks may have evolved to have this property.^{6,9} Recently, a few papers have questioned whether this property is an artifact of biases that are inherently present in the experimental screens that are performed to detect the protein interactions. Researchers have also showed that protein interaction networks have the small world property,⁹ i.e., the average distance between any two vertices in the graph is small. A number of graph-theoretic properties of protein interaction networks that have been studied so far, focus on the local properties of the graph, e.g., degree distribution, Centroid value, excentricity,¹⁰ clustering coefficient¹¹ and average distance¹¹ between any two vertices in the network.

In this thesis, we examine the betweenness centrality of protein interaction networks. The vertex betweenness centrality of a given vertex is the fraction of shortest paths, counted over all pairs of vertices, that pass through that vertex. Edge betweenness centrality is similarly defined for an edge. Since these measures consider both the local and the global structure of the graph, we believe they may be more appropriate for studying biological networks, in particular, and complex networks in general.

1.2 Contributions Of This Thesis

In this thesis we apply the graph-theoretic concept of betweenness centrality to the protein interaction networks of three organisms. These networks contain both physical and genetic interactions. We observe that the vertex betweenness distribution follows a power law for all the networks. We also observe that the distribution of the edge betweenness centrality has a Poisson-like distribution with a very sharp spike for all three networks. We further analyse this behavior by constructing random graphs with the same degree distribution as the protein interaction networks of the three organisms, and computing the edge betweenness distribution of the random graphs. We observe that the edge betweenness distribution for the random graphs also possess a Poisson-like distribution with a very sharp spike.

We also construct random graphs of different sizes and whose degree distribution follows a power law with different values of the power law exponent. We observe that the edge betweenness distribution for these random graphs also possess a Poisson-like distribution with a very sharp spike. We conjecture at the end of our experiments that graphs whose degree distribution follows a power law have an edge betweenness distribution similar to the ones we observe for protein interaction networks.

1.3 Readers Guide

The rest of the thesis is organised as follows. Chapter 2 describes the experiments used to generate the protein interaction data, the interaction datasets currently available, and previous results on the properties of biological networks. Chapter 3 describes different graph models used to generate complex networks such as protein interaction networks. Chapter 4 defines betweenness centrality and the applications of betweenness centrality and chapter 5 describes the algorithms we have implemented for computing the vertex betweenness of all the vertices and the edge betweenness of all the edges in a graph. Chapter 6 shows the results observed from the experiments. Chapter 7 gives the conclusion of the study and discusses future research directions in this area.

Chapter 2

Properties Of Biological Networks

Biological networks have been mainly constructed from experiments or from scientific literature. These networks contain interactions between proteins, DNA, RNA, metabolites, etc. There are many publicly available databases of these interaction data. Many properties of these networks have already been studied and these studies have yielded many important results. In the following sections, we describe the different experiments that generate the protein interaction network data, the biological datasets that are available and the previous results on the properties of biological networks.

2.1 Biological Experiments

In this section, we describe the biological experiments used to generate protein-protein interactions and other types of interactions. Two of the experiments, the two-hybrid system and affinity precipitation, produce *physical interaction* data. In a physical interaction between two protein, the proteins actually interact in the cell. Synthetic lethality experiments produce *genetic interaction* data. In genetic interactions, the proteins do not physically interact with each other in the cell.

2.1.1 The Two Hybrid System

The two hybrid system is used to detect if one protein (prey) physically interacts with another (bait). This system detects the interaction between the bait and prey by using a transcription factor (usually GAL4) to promote the transcription of a reporter gene.¹² Transcription factors such as GAL4 have distinct structural and functional units called domains. The GAL4 protein has a DNA Binding Domain (DBD) as well as an Activation Domain (AD). When GAL4 binds to its cognate binding site, the activation domain is brought close to the promoter, allowing the activation domain to interact with the transcriptional machinery and resulting in activation of transcription. A reporter gene is used to detect the activation of transcription.

The above machinery is used in the two hybrid system as follows:

- **DBD Hybrid:** This hybrid contains the DBD fused to the bait. This fusion protein can bind to the DNA, but cannot activate transcription because the bait does not contain an activation function.
- **AD Hybrid:** This hybrid contains the AD fused to the prey. Usually, a recombinant DNA “library” is prepared in which genes for many different proteins are fused to the AD. Then both hybrid proteins are expressed in the same cell. Those AD hybrids expressing the reporter gene are identified and purified for further characterization.

Typically, libraries containing large numbers of different proteins are screened against one bait and all the proteins (prey) expressing the reporter gene, are the few proteins that can interact with the bait.^{13,14} The two hybrid system has been used to generate protein-protein interaction data for the following organisms, baker's yeast,^{13,14} fruit fly¹⁵ and worm.¹⁶

2.1.2 Affinity Precipitation

Affinity precipitation is another method used to find interacting proteins. Here the interacting proteins are found by analysing all the proteins that are bound to the bait protein. The bait protein is first tagged with an antibody binding tag. To separate the bait protein from the rest of the proteins in the cell, an antibody is added which binds itself to the antibody binding tag. Another protein (the antigen)

that binds to the antibody is then used to extract the antibody, the antibody tag, bait protein and any the proteins bound to the bait protein. The complex obtained is separated by gel electrophoresis and further analysed using mass spectroscopy to identify the interacting proteins. This method has been used to detect a large number of protein complexes in Yeast and thus has aided in the formation of protein interaction networks.^{17,18}

2.1.3 Synthetic Lethality

There exists a *synthetically lethal* interaction between two genes if the mutation in each gene by itself is not lethal to the cell, but the combination of both mutations causes cell death. These type of interactions have been found between two genes in the same biochemical pathway as well as between two genes in different biochemical pathways. Genes involved in many cellular processes have also been identified using synthetic lethal screens. Tong *et al.* have developed the Synthetic Genetic Array (SGA) analysis to do multiple screens and detect a large number of synthetic lethal interactions.¹⁹

2.2 Biological Datasets

In this section we describe the various protein interaction datasets that are currently available.

2.2.1 BIND: Biomolecular Interaction Network Database

The Biomolecular Interaction Network Database (BIND)²⁰ is a collection of records documenting molecular interactions. BIND contains interaction data for a variety of organisms like mouse, yeast, HIV virus etc. The interaction data has been obtained from high-throughput data submissions and hand curated information from scientific literature.

2.2.2 DIP: Database of Interacting Proteins

The DIP database²¹ contains experimentally determined interactions between proteins for a large number of organisms like human, yeast, mouse, worm etc. The interactions in the DIP database have been generated by combining the information from a variety of sources. The data stored was curated manually as well as using computational approaches.

2.2.3 GRID: The General Repository for Interaction Datasets

The GRID datasets²² are available for yeast, fly and worm. GRID contains physical, genetic and functional interactions between proteins. The data has been generated from biological experiments such as the two hybrid system, affinity precipitation, and synthetic lethality. Yeast network has 4920 vertices and 17816 edges. Fly network has 7940 vertices and 25665 edges. Worm network has 2803 vertices and

4371 edges.

2.3 Previous Results

One of the first observed properties of biological networks has been that the degree distribution of biological networks follows a power law. It has been observed that in metabolic networks, there are few metabolites such as pyruvate and coenzymeA, which participate in a number of reactions and function as hubs.⁷ The scale free behaviour and the presence of hubs are also seen in the protein interaction network of yeast.^{13,14} The small world property is another feature of biological networks.⁹ Different properties of the interaction data have been used for functional annotation of proteins and genes.¹⁻⁵ The structure of protein interaction data has been used for predicting new interactions between existing proteins.²³ The clustering coefficient of a vertex measures how connected the neighbours of that vertex are to each other. The average clustering coefficient of a graph is the average of the clustering coefficient measure of all the vertices in the graph. The average clustering coefficient measures the overall tendency of vertices in the graph to form clusters. The average clustering coefficient of biological networks is observed to be high.²⁴ Motifs are certain subgraphs that are over represented in the real graph, when compared to a randomized version of the same network. The high degree of evolutionary conservation of motif constituents with in the yeast protein interaction network

indicates that motifs are of direct biological relevance.²⁵

Chapter 3

Graph Generation Models

A biological network such as a protein-protein interaction or a synthetic lethal interaction network can be represented as an undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. Vertices in such a graph represent proteins while edges represent interactions between the proteins. Let G have n vertices and m edges. Let N_v be the set of neighbours of a vertex v . Let $d_v = |N_v|$ be the degree of vertex v . Researchers have observed that networks such as the Internet, citation patterns in scientific papers and some biological networks are scale free.⁶ A network is *scale-free*, if its degree distribution follows a power law i.e., the probability $P(k)$ that a vertex in the network has degree k is proportional to $k^{-\gamma}$ for some $\gamma > 0$. A graph generation model is an algorithm or certain steps to follow, so that we may generate graphs with certain properties. The different graph generation models suggested for complex networks such as biological networks that

show the scale-free property have been described in the following sections.

3.1 Erdős-Rényi Model

Although the Erdős-Rényi model was not initially proposed to explain the evolution or structure of biological networks, we include it since it is a well studied model for the generation of random graphs. An Erdős-Rényi graph $G(n, p)$ is a graph with n vertices such that the probability of having an edge (u, v) in G is p for any vertices u and v in G .²⁶

Although the Erdős-Rényi model is a well studied model, it does not fully capture the features exhibited by biological networks. The presence of many highly connected hubs is a feature that is observed in biological networks. An Erdős-Rényi graph is unlikely to have such a property, since the probability of occurrence of every edge is the same.

3.2 Barabási Albert Model

Barabási and Albert⁶ conjecture that the scale free property of complex networks arises because:

1. the network grows with the addition of more vertices and

2. a new vertex preferentially attaches itself to vertices with high degree.

The propose a growth based model²⁷ in which a new vertex is created at each time step and the newly arrived vertex preferentially attaches itself to existing vertices with higher degree. Therefore in this case vertices with higher degree have a higher probability of connecting to the new vertex. The probability p_v of creating an edge between an existing vertex v and the newly added vertex is

$$p_v = (d_v + 1)/(|E| + |V|)$$

where $|E|$ and $|V|$ are, respectively, the number of edges and vertices currently in the network (counting neither the new vertex nor the other edges that it is incident on).

Due to preferential attachment, a vertex with a higher degree will continue to increase its connectivity at a higher rate, this does explain the presence of hubs in such networks.

3.3 Watts Strogatz Small World Model

The small world model is a graph generating model proposed by Watts and Strogatz.⁹ Graphs which have the small world property have low characteristic path lengths i.e., the average distance between any two vertices in the graph is small and also high clustering coefficient. The algorithm to generate a graph takes as input a

regular graph with n vertices with k edges incident on each vertex and a probability p . The algorithm chooses an edge at random with probability p , then one of the end points of the edge is changed to another vertex, again chosen at random.

3.4 Eppstein Wang Model

Eppstein and Wang²⁸ proposed a steady state method for generating scale-free networks. A steady state model is not a growth based model i.e., the model does not involve the addition of new vertices or edges. The input to the algorithm is the number of edges m , the number of vertices n and a model parameter r . The model starts by generating a graph with n vertices and m edges, by randomly adding edges between the vertices until there are m edges. The algorithm then modifies the initial graph by executing the following sequence of steps r times:

1. Pick a vertex v at random. Repeat this step until $d_v > 0$.
2. Pick an edge $(u, v) \in G$ at random.
3. Pick a vertex x at random.
4. Pick a vertex y proportional to degree of y .
5. If (x, y) is not an edge and if x is not y , then add edge (x, y) to G and remove edge (u, v) from G .

This is a simple model for generating scale-free networks, because it produces a power-law graph without the addition of extra vertices and edges, by evolving the existing graph while maintaining the same number of edges and vertices. Eppstein and Wang simulated the model on graphs with different sizes and different densities, where $density = m/n$. Each simulation was performed five times and the model parameter r was chosen to be 10^7 . The degree distribution was observed to converge to a power law distribution as the value of r increased, for many sizes and densities of the graph.

Chapter 4

Betweenness Centrality

In this chapter, we define the graph-theoretic concept of betweenness centrality, which is central to this thesis. This concept takes into account the global as well as the local features of a network. We present many applications of betweenness centrality and describe the two algorithms, one for computing the vertex betweenness centrality and the other for computing edge betweenness centrality, for all the vertices and edges in the graph.

4.1 Definition

In the context of social networks, Freeman²⁹ defined a number of measures of centrality to find out how influential a person or group is.³⁰ In this thesis, we are

concerned with a measure called betweenness centrality. The *vertex betweenness centrality* $BC(v)$ of a vertex $v \in V$ is the sum over all pairs of vertices $u, w \in V$, of the fraction of shortest paths between u and w that pass through v :

$$BC(v) = \sum_{\substack{u, w \in V \\ u \neq w \neq v}} \frac{\sigma_{uw}(v)}{\sigma_{uw}}$$

where $\sigma_{uw}(v)$ denotes the total number of shortest path between u and w that pass through vertex v and σ_{uw} denotes the total number of shortest paths between u and w .

Similarly the *edge betweenness centrality* $BC(e)$ of an edge $e \in E$ is defined as the sum over all pairs of vertices (u, w) , of the fraction of shortest paths between u and w that passes through e .

$$BC(e) = \sum_{\substack{u, w \in V \\ u \neq w}} \frac{\sigma_{uw}(e)}{\sigma_{uw}}$$

where $\sigma_{uw}(e)$ denotes the total number of shortest path between u and w that pass through edge e and σ_{uw} denotes the total number of shortest paths between u and w .

4.2 Applications

Since its formulation, the betweenness centrality measure has been used in a variety of settings. Betweenness centrality measures were obtained for a disease outbreak network to investigate a tuberculosis outbreak and methods of disease control.³¹

Similarly betweenness centrality has been used in variety of settings to infer useful information about the network. In the following sections, we describe the application of betweenness centrality measures to classify scale free networks, measure the susceptibility of complex networks and to find communities in networks.

4.2.1 Classification of scale free networks

Betweenness centrality has been equated to “load”,³² the amount of traffic a vertex or edge has to handle in a network such as the Internet, when every pair of vertices is sending and receiving packets along the shortest paths connecting the pair. Goh, Kahng and Kim showed that the load distribution of scale free networks followed a power law with power law exponent $\delta \sim 2.2$, for different values of γ , where γ is the power law exponent of the network.³² Goh *et al* later concluded that scale free networks could be classified using the power law exponent of the betweenness centrality distribution of the network³³ δ , where the value of δ could be 2 or 2.2 for different networks. Kim, Noh and Jeong also studied the betweenness centrality distribution of scale free trees, and showed that scale free networks can be classified

by the power law exponent of the betweenness centrality distribution.³⁴

4.2.2 Susceptibility of complex networks

Holme and Kim studied the susceptibility of complex networks like the Internet to fragmentation, due to vertex or edge overloading.^{35,36} Load in this context has the same definition as in the previous section. The implication of this study is on real-world communication networks such as the Internet, which are constantly growing and evolving. Load here is characterized by betweenness centrality. Edge or vertex overloading occurs, when the vertex or edge has more network traffic than it can handle or has the capacity for. For finding the effect of overloading on evolving scale free networks, it was defined in the experiments that were conducted, that overloading occurred when the betweenness centrality of an edge or a vertex exceeded a maximum set value. Overloading of edges and vertices in a communication network leads to the vertex or edge being shutdown and therefore leads to breakdown avalanches in the network.^{35,36} When an edge or a vertex shuts down due to overloading, this increases the load on the other edges and vertices in the graph, and thus may lead to overloading of more vertices and edges. This is known as a breakdown avalanche. Breakdown avalanches lead to further fragmentation of the network. This study was conducted on scale free networks generated using the Barabási Albert graph generation model.²⁷ Holme *et al* also studied the different reactions of the complex network to different attack strategies.³⁷ In these

experiments, instead of removing overloaded edges and vertices, certain procedures were followed such as removing vertices with high degree, removing vertices with high betweenness etc to observe the fragmentation in the network. The attack strategy based on recalculated betweenness centrality was found to be the most harmful i.e., remove the vertex with the highest value of betweenness centrality, then recalculate the betweenness centrality measure for all the vertices in the graph and repeat the procedure of removing the vertex with the highest value of betweenness centrality measure. High correlation between the degree and the betweenness centrality of vertices was also observed for complex networks.³⁷

4.2.3 Communities in Networks

Communities in networks are groups of tightly knit vertices, which are joined by looser connections. Girvan and Newman proposed an algorithm based on edge betweenness for finding communities in networks.³⁸ The algorithm works as follows:

1. Find the edge betweenness of all the edges in the network.
2. Remove the edge with the highest betweenness value from the network.
3. Recalculate the edge betweenness values for all the edges in the remaining network.
4. Return to step 2 until the graph has no edges.

Holme, Huss and Jeong modified the above algorithm to detect sub-networks in biological networks (metabolic networks).³⁹ They represented the biological network as a bipartite graph and the reactions between the different metabolites as reaction vertices. They defined the effective betweenness of a reaction vertex to be the vertex betweenness divided by the indegree of the vertex. After calculating the effective vertex betweenness of every vertex in the network, the reaction vertex with the highest effective betweenness value was removed and the process was repeated until there were no reaction vertices. This process broke down the biological network into communities. Radichhi et al also modified the Girvan Newman algorithm³⁸ to find communities in networks.⁴⁰

Modified versions of the Girvan Newman edge betweenness clustering algorithm³⁸ have also been used to find communities in gene networks.⁴¹ In these networks, two genes are connected if they appeared together in scientific literature. It was observed that genes in the same community had the same function, and genes in different communities had different functions.⁴¹ Recently the edge betweenness clustering algorithm has been applied to protein interaction networks,⁴² the authors report that the algorithm yields clusters of proteins which have similar functions.

The betweenness centrality distribution of vertices in yeast protein interaction networks have also been studied.⁴³ Existence of vertices with high betweenness values and low degree were observed. The vertices of high betweenness measure were found to be essential proteins, essential protein are those which are necessary

for proper cell functioning and the lack of which in the cell, can lead to cell death.

Chapter 5

Algorithms For Computing Betweenness Centrality

5.1 Background

In this chapter, we describe the algorithm we have implemented for computing the vertex and edge betweenness centrality measures of all the vertices and edges in a graph. Given a graph $G = (V, E)$ with n vertices and m edges, let ω be the weight function on the edges of the graph. Therefore, for unweighted graphs $\omega(e) = 1$, for $e \in E$. Define a path from $s \in V$ to $t \in V$ to be a sequence of vertices such that the path starts at s and ends at t , and there is an edge in G connecting each vertex in the path to its successor in the path. The length of the path is the sum of the weights of the

edges in the path; for an unweighted graph, the length of the path is the total number of edges in the path. Let $d_G(s, t)$ denote the minimum length of any path connecting s and t in G . By definition, $d_G(s, s) = 0$ and $d_G(s, t) = d_G(t, s)$. A vertex $v \in V$ lies on a shortest path between $s, t \in V$, if and only if $d_G(s, t) = d_G(s, v) + d_G(v, t)$.

Let σ_{st} denote the total number of shortest paths between vertices s and t and $\sigma_{st}(v)$ denote the total number of shortest paths between vertices s and t that pass through v , where $s, t, v \in V$. Note that $\sigma_{st} = \sigma_{ts}$ and $\sigma_{st}(v) = \sigma_{ts}(v)$. Then the betweenness centrality measure²⁹ for a vertex $v \in V$ is

$$BC(v) = \sum_{\substack{s, t \in V \\ s \neq t \neq v}} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (5.1)$$

A fundamental component of all the algorithms we discuss is a procedure to count the number of shortest paths from a given vertex $s \in V$ to all the other vertices in G . We do so by implementing Dijkstra's shortest path algorithm to compute the shortest path directed acyclic graph (DAG) D_s rooted at s rather than the shortest path tree T_s rooted at s . We define D_s as follows: A node v is a parent of node t in D_s , if v lies on a shortest path from s to t . Note that we can augment the shortest path tree T_s rooted at s into D_s in $O(n + m)$ by using the following observation: let $d_G(s, t)$ be the length of the path between s and t in T_s . For an edge $e = (v, t) \in E$, if $d_G(s, t) = d_G(s, v) + \omega(e)$, then v is a parent of t in D_s . We define $P_s(t)$ as the set of parents of t in D_s .

$$P_s(t) = \{v \in V : (v, t) \in E, d_G(s, t) = d_G(s, v) + \omega(v, t)\}$$

Given D_s , we can calculate σ_{st} , for every node $t \in V$ as follows:

$$\sigma_{st} = \sum_{v \in P_s(t)} \sigma_{sv}$$

We now sketch a naive algorithm for computing the vertex betweenness centrality for every node in G . The algorithm has the following steps:

1. For every node $s \in V$
 - (a) Compute D_s and σ_{st} for every node $t \in V$.
 - (b) For every node $v \in V, v \neq s$
 - i. Delete v from G . Let G' be the resulting graph.
 - ii. Compute D'_s , the shortest path DAG rooted at s in G' .
 - iii. For a node $t \in G'$, let σ'_{st} be the number of shortest paths from s to t in G' .
 - iv. Set $\sigma_{st}(v) = \sigma_{st} - \sigma'_{st}$ for all $t \in G'$.
2. For every node $v \in V$ set

$$BC(v) = \sum_{\substack{s, t \in V \\ s \neq t \neq v}} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Since this algorithm involves Dijkstra's shortest path algorithm $O(n^2)$ times, its running time is $O(n^2(n + m) \log n)$. In this thesis, we use the more efficient algorithms devised by Brandes⁴⁴ and Newman¹¹ for computing the betweenness centralities of all the vertices and of all the edges respectively, in the graph.

5.2 Brandes Vertex Betweenness Algorithm

Brandes⁴⁴ developed a more efficient algorithm than the one described above by noting that it is not necessary to invoke the shortest path algorithm $O(n^2)$ times to compute the betweenness centrality of all the vertices in G . The $O(n)$ shortest path DAGs rooted at each node of G contain all the required information.

Brandes defines the *pair-dependency* $\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}$, where $s, t, v \in V$. Clearly

$$BC(v) = \sum_{s,t \in V} \delta_{st}(v)$$

Therefore given the pairwise distances and also the number of shortest paths, we can calculate for a pair $s, t \in V$ and a vertex $v \in V$, the pair dependency $\delta_{st}(v)$.

Therefore betweenness centrality is usually calculated in two steps:

1. Compute the length and number of shortest paths between all pairs of vertices.
2. Sum all pair-dependencies.

Brandes defines the *dependency* of a vertex $s \in V$ on a vertex $v \in V$ as

$$\delta_{s\bullet}(v) = \sum_{t \in V} \delta_{st}(v)$$

By (5.1), we have

$$BC(v) = \sum_{s \in V} \delta_{s\bullet}(v) \quad (5.2)$$

Brandes proves the following recursive relation on $\delta_{s\bullet}(v)$, which is crucial to his algorithm:

$$\delta_{s\bullet}(v) = \sum_{w|v \in P_s(w)} \frac{\sigma_{sv}}{\sigma_{sw}} \cdot (1 + \delta_{s\bullet}(w)) \quad (5.3)$$

Therefore, given D_s , we can calculate $\delta_{s\bullet}(v)$ for all the vertices $v \in V$ by a traversal of D_s in topological order. We can now describe Brandes algorithm completely:

1. For every vertex $s \in V$
 - (a) Compute D_s and σ_{st} for all $t \in V$.
 - (b) Using (5.3) compute the dependency of s on every other vertex in the graph.
2. Compute $BC(v)$ for all $v \in V$, using (5.2).

For each node $s \in V$, step (a) takes $O((n + m) \log n)$ time and step (b) takes $O(n + m)$ time. Therefore, the total time spent by the algorithm is $O(n(n + m) \log n)$. The space used by the algorithm is $O(n + m)$. Note that if G is unweighted, we can use Breadth First Search instead of Dijkstra's shortest path algorithm, reducing the running time to $O(n(n + m))$.

5.3 Newman Edge Betweenness Algorithm

The notion of edge betweenness is based on the number of shortest paths that pass through a certain edge. The *edge betweenness* $BC(e)$ for an edge $e \in E$ is given by

$$BC(e) = \sum_{\substack{s, t \in V \\ s \neq t}} \frac{\sigma_{st}(e)}{\sigma_{st}} \quad (5.4)$$

where $\sigma_{st}(e)$ is the number of shortest paths from vertex $s \in V$ to vertex $t \in V$ that pass through edge $e \in E$.

We describe Newman's algorithm using the notation we have developed earlier. Let's define *pair-dependency* on an edge $e \in E$ as $\delta_{st}(e) = \frac{\sigma_{st}(e)}{\sigma_{st}}$, where $s, t \in V$. Note that $\delta_{st}(e) = 0$, if e is not an edge in the D_s . We define a *dependency* of a vertex $s \in V$ on an edge $e \in E$ as

$$\delta_{s\bullet}(e) = \sum_{t \in V} \delta_{st}(e)$$

Clearly,

$$BC(e) = \sum_{s \in V} \delta_{s\bullet}(e) \quad (5.5)$$

Let u and v be the vertices connected by e . Assume without loss of generality that $u \in P_s(v)$, i.e., at least one shortest path from s to v passes through u . Define the set of predecessors of e in D_s : $P_s(e)$ as the set of all edges incident on u in D_s . Newman proves the following recursive relation:

$$\delta_{s\bullet}(e) = \sum_{w|e \in P_s(w)} \frac{\sigma_{su}}{\sigma_{sv}} \cdot (1 + \delta_{s\bullet}(w)) \quad (5.6)$$

It is easy to modify the Brandes algorithm to use this recurrence relation (5.6) to calculate $BC(e)$ for all edges $e \in E$. The algorithm runs in $O(n(m+n))$ time for unweighted networks.

Chapter 6

Results

In this thesis, we analysed the genetic and physical interaction networks of the following organisms: *Saccharomyces cerevisiae* (yeast), *Caenorhabditis elegans* (worm) and *Drosophila melanogaster* (fly). We obtained these data sets from the General Repository for Interaction Datasets²² (GRID). GRID is a comprehensive database of genetic and physical interactions in *Saccharomyces cerevisiae* (yeast), *Caenorhabditis elegans* (worm) and *Drosophila melanogaster* (fly). The yeast dataset had physical interactions from affinity precipitation and two-hybrid experiments,^{13,14} and genetic interactions from synthetic lethality experiments.¹⁹ The yeast interaction network has 4920 vertices and 17816 edges. The fly dataset had interactions from two-hybrid experiments and genetic interactions. The fly interaction network has 7940 vertices and 25665 edges. The worm interaction network has 2803 vertices and 4371 edges, and has interactions detected by

two-hybrid experiments.

We first computed the vertex betweenness distribution for all three networks, and observed that it follows a power law. We also studied the vertex betweenness vs. degree correlation for all three networks. In the edge betweenness distribution for all the three networks, we saw a strange behaviour, i.e., presence of a large fraction of edges with the same betweenness value. To uncover the reason behind this behavior, we generated random graphs with the same degree distribution as the original networks. We also generated random graphs with different densities and whose degree distribution followed power law with different values of the power law exponent. We plotted the average edge betweenness distribution for all these graphs too.

The values of edge betweenness for the edges in a graph were normalised by dividing it by the total number of edges in the graph. This was done so that we may compare graphs with different sizes, i.e., compare graphs with different number of nodes and edges.

6.1 Vertex Betweenness

Since vertex betweenness took into account the fraction of the number of shortest paths that pass through a vertex over all pair of vertices, we initially wanted to check if the vertex betweenness value for a vertex in a biological network, would

allow us to predict how lethal a gene is or if the protein was essential etc. We calculated the vertex betweenness values using the Brandes algorithm.

6.1.1 Vertex Betweenness Distribution

We applied the Brandes Algorithm⁴⁴ to find the vertex betweenness values for all three networks. We wanted to view the vertex betweenness distribution as well as the correlation of betweenness centrality of a vertex with its degree.

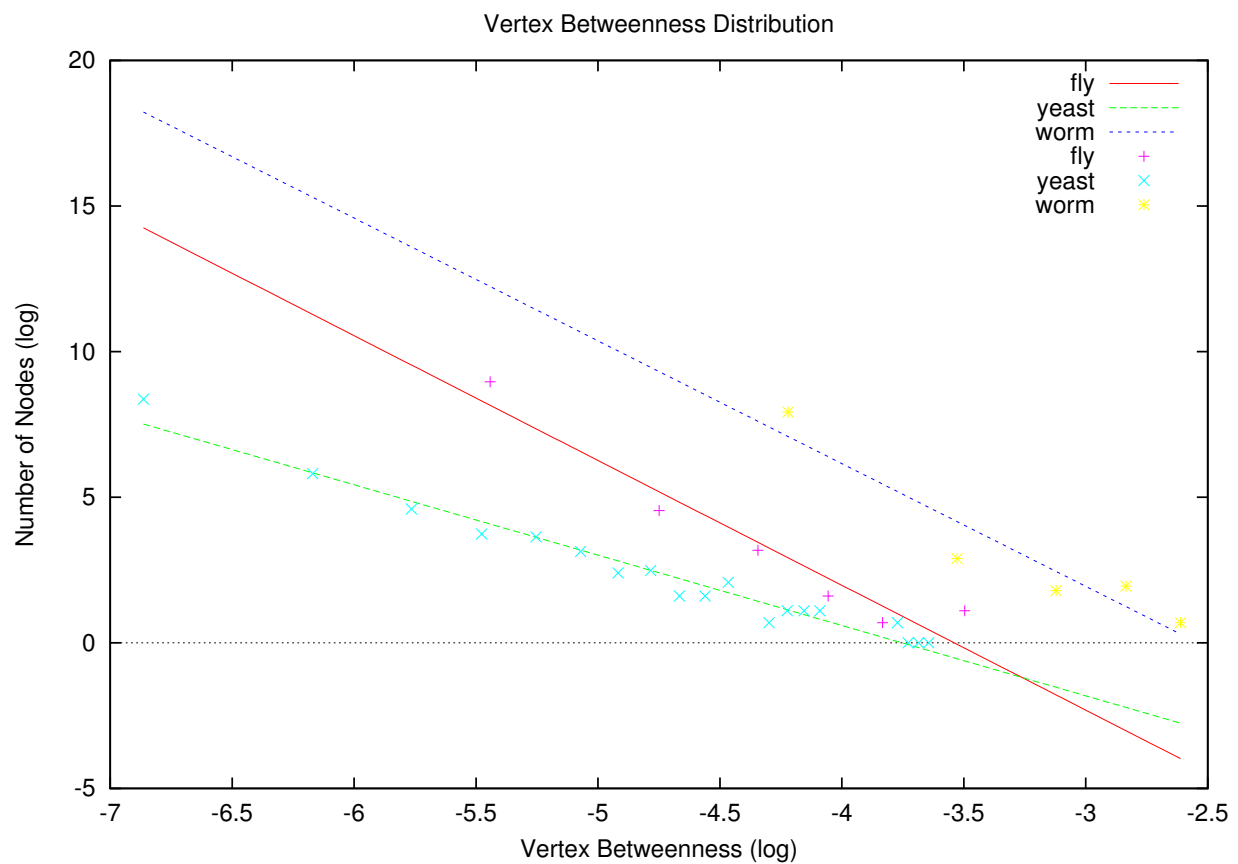


Figure 6.1: Vertex betweenness distribution for yeast, fly and worm interaction data.

Figure 6.1 shows that there are large number of vertices with vertex betweenness in a certain range or nearly same value. Figure 6.1 shows the log-log plot of vertex betweenness distribution for all the the three networks. The value for x-intercept is -3.5, the y-intercept is -15 and the slope is -4.2 for the power law fit for the fly network. The value for x-intercept is -3.7, the y-intercept is -9 and the slope is -2.4 for the power law fit for the yeast network. The value for x-intercept is -2.5, the y-intercept is -11 and the slope is -4.4 for the power law fit for the worm network. From the results in figure 6.1 it is clear that the vertex betweenness distribution exhibits a power law for the three networks.

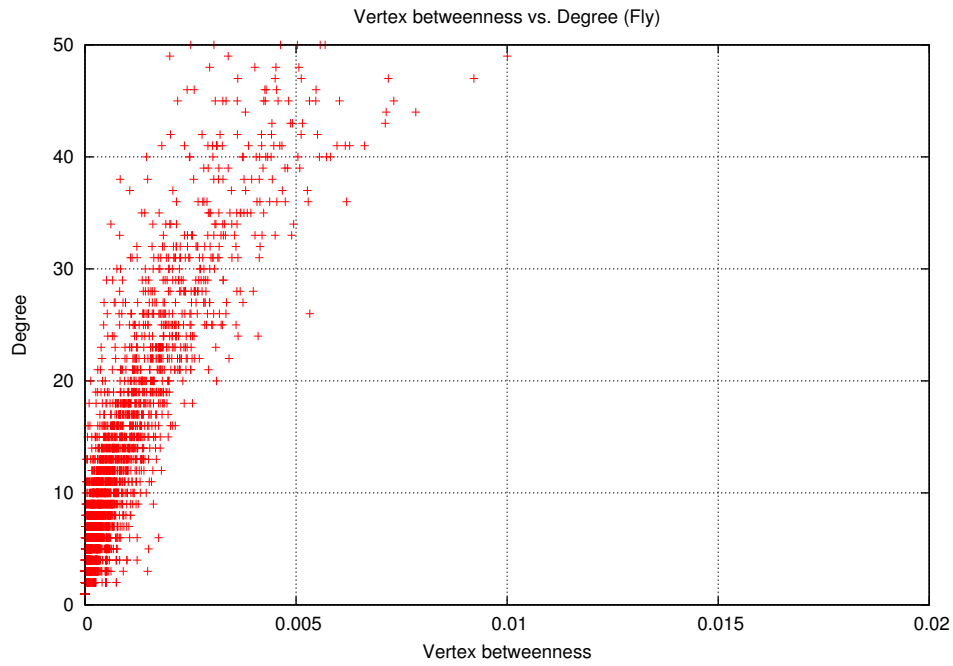
6.1.2 Vertex Betweenness vs. Degree Correlation

Each point in the plots in Figure 6.2 and Figure 6.3 represents the betweenness centrality and the degree of a single vertex.

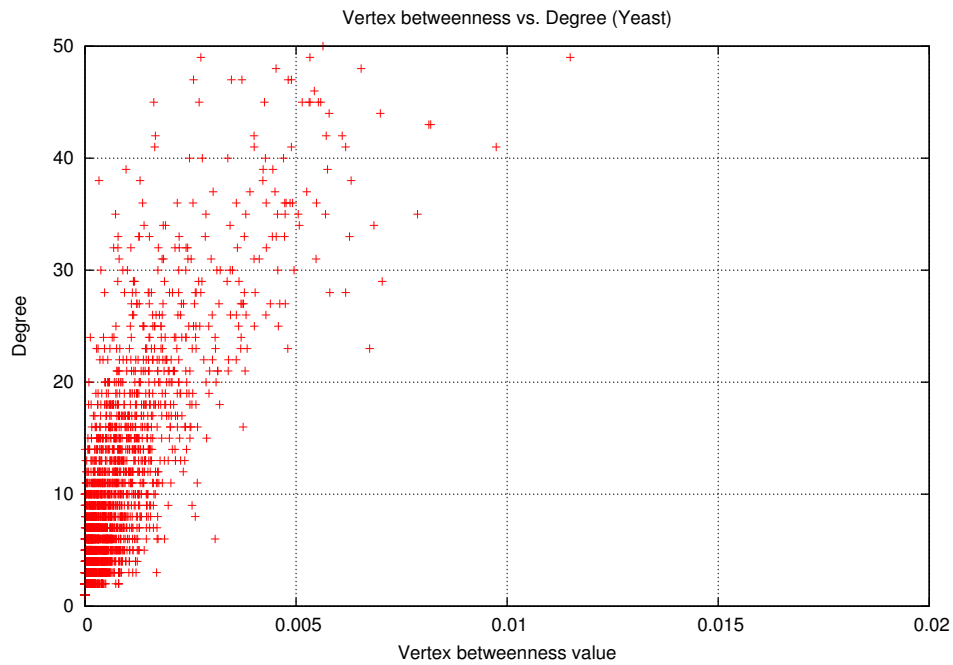
To create the plot in figure 6.4, we binned the vertices of each network by degree.

Bin i , $i > 0$, contained all the vertices with degree between i and $i - 10$. For each bin i , we plot the degree and the mean of the betweenness centrality values of the vertices in that bin.

From the figures, figure 6.2, figure 6.3 and figure 6.4, it is clear that vertex betweenness values have high correlation with degree of a vertex i.e. higher the degree of the a vertex, higher its vertex betweenness value will be.



(a) The fly network



(b) The yeast network

Figure 6.2: Vertex betweenness value vs. Degree for each vertex.

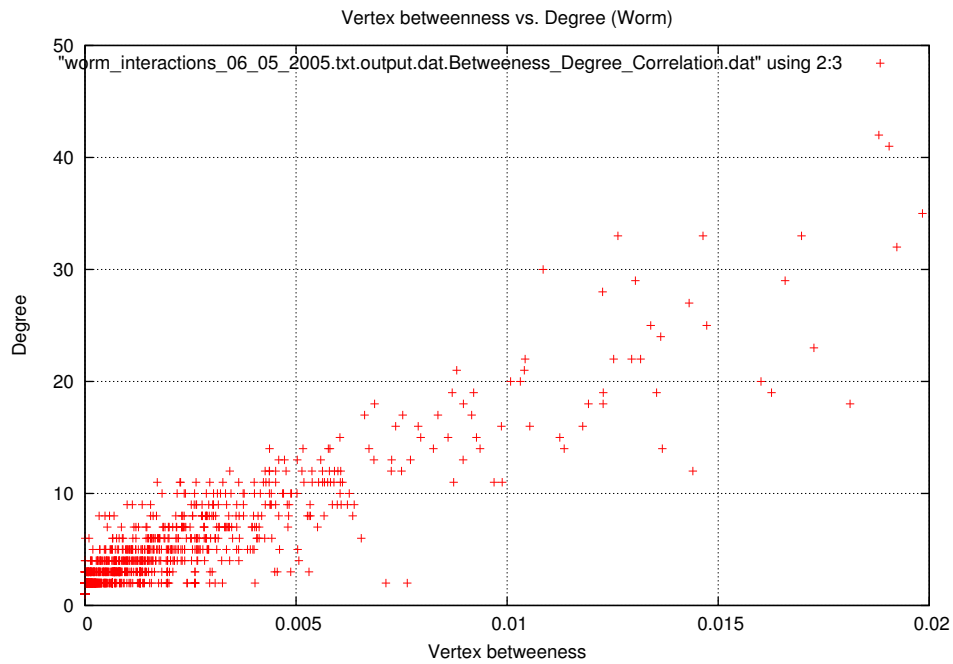


Figure 6.3: Vertex betweenness value vs. Degree for each vertex of the worm network.

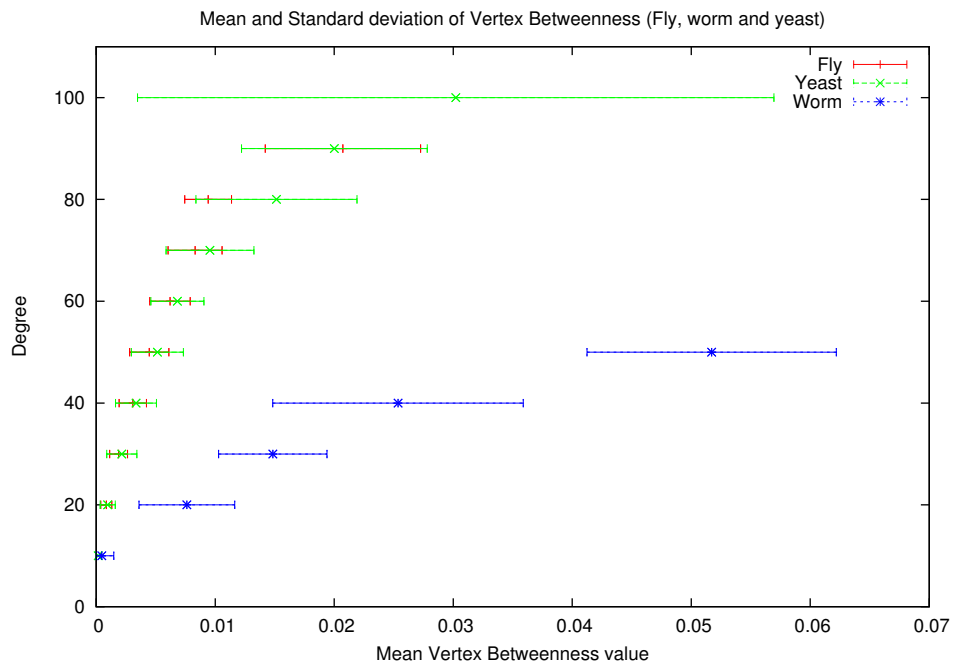


Figure 6.4: Mean and standard deviation of vertex betweenness values of vertices having degrees in a certain range vs The maximum degree of the range.

6.2 Edge Betweenness

Edge betweenness takes into account the fraction of shortest paths between two vertices that pass through an edge, over all pair of vertices. We expected the edge betweenness distribution for all three networks to follow power law, but we were surprised to discover that, not only did it not follow a power law, but it had a very strange shape.

6.2.1 Edge Betweenness Distribution

We computed the edge betweenness values for all the edges in the network using the Newman algorithm¹¹ for all the three interaction networks. The edge betweenness distribution is given in figure 6.5 and 6.6 for all the three interaction networks: In these plots, for each range of edge betweenness values (we used a thousand bins), we plot the fraction of edges with edge betweenness value in that range.

From figures 6.5, 6.6 and 6.7, it is very interesting to see a sudden increase in the number of edges with a certain edge betweenness value in the edge betweenness distribution of all three of the datasets. The larger spike is also followed by a smaller one in all three figures. The spike signifies the fact that there are a large number of edges with nearly the same edge betweenness value. In figure 6.7, we simultaneously plot the individual distributions displayed in figures 6.5 and 6.6. It is also interesting to see that, when the edge betweenness distribution is normalized

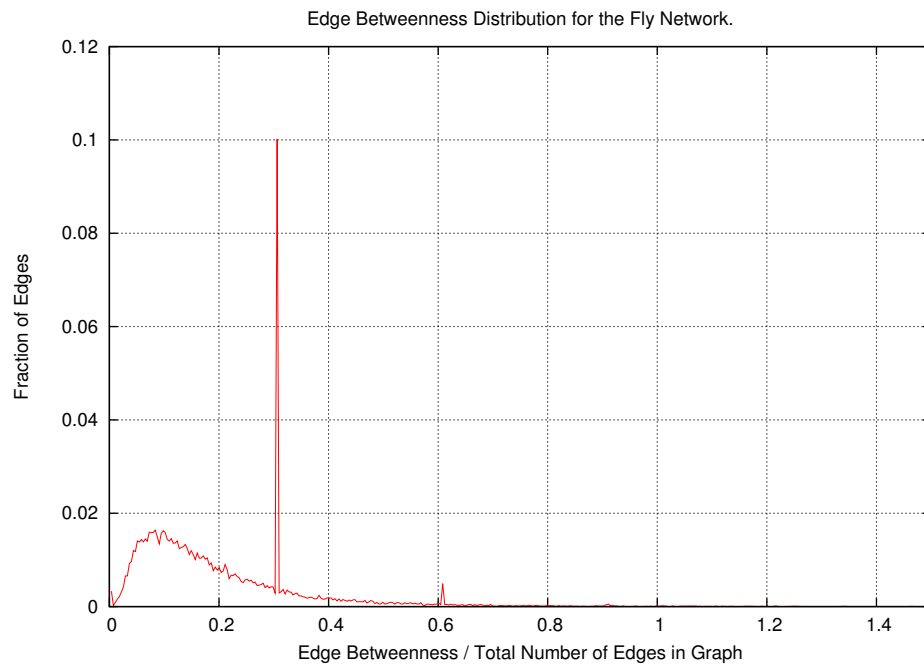
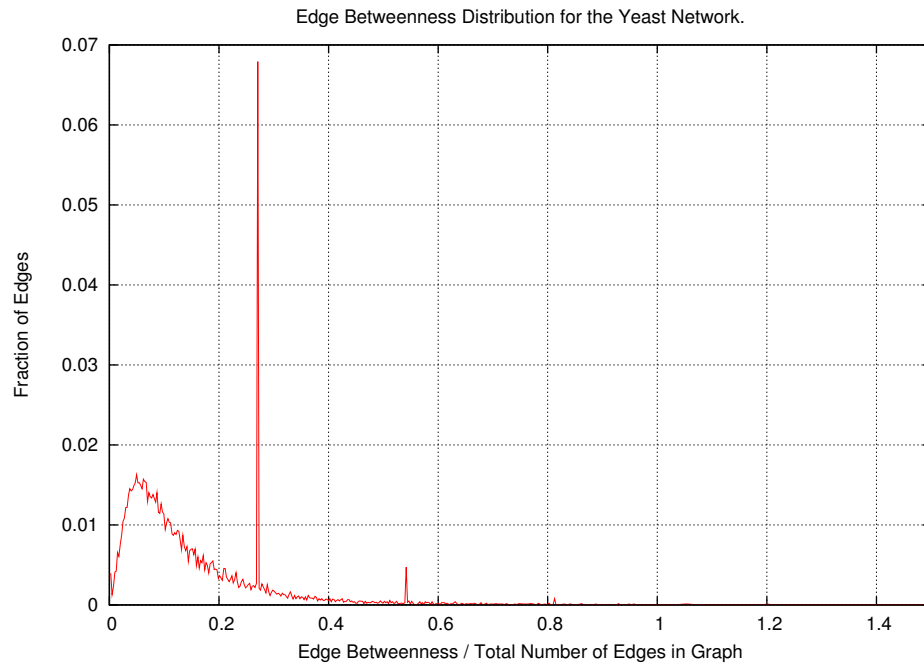
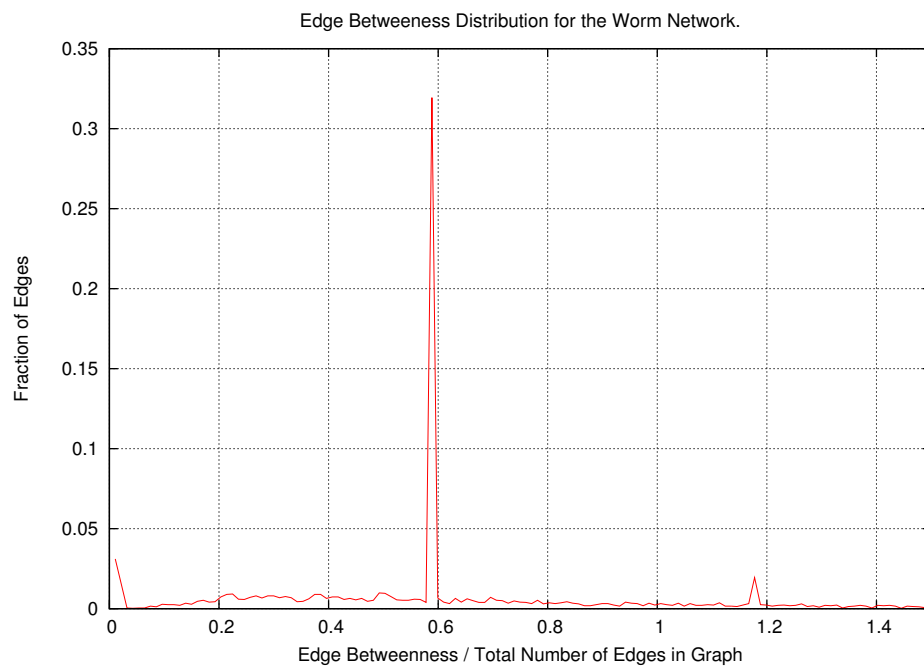


Figure 6.5: Edge betweenness distribution for the fly network. We divide each edge betweenness value by the total number of edges in the graph.



(a) Yeast



(b) Worm

Figure 6.6: Edge betweenness distribution for the yeast and worm networks. We divide each edge betweenness value by the total number of edges in the graph.

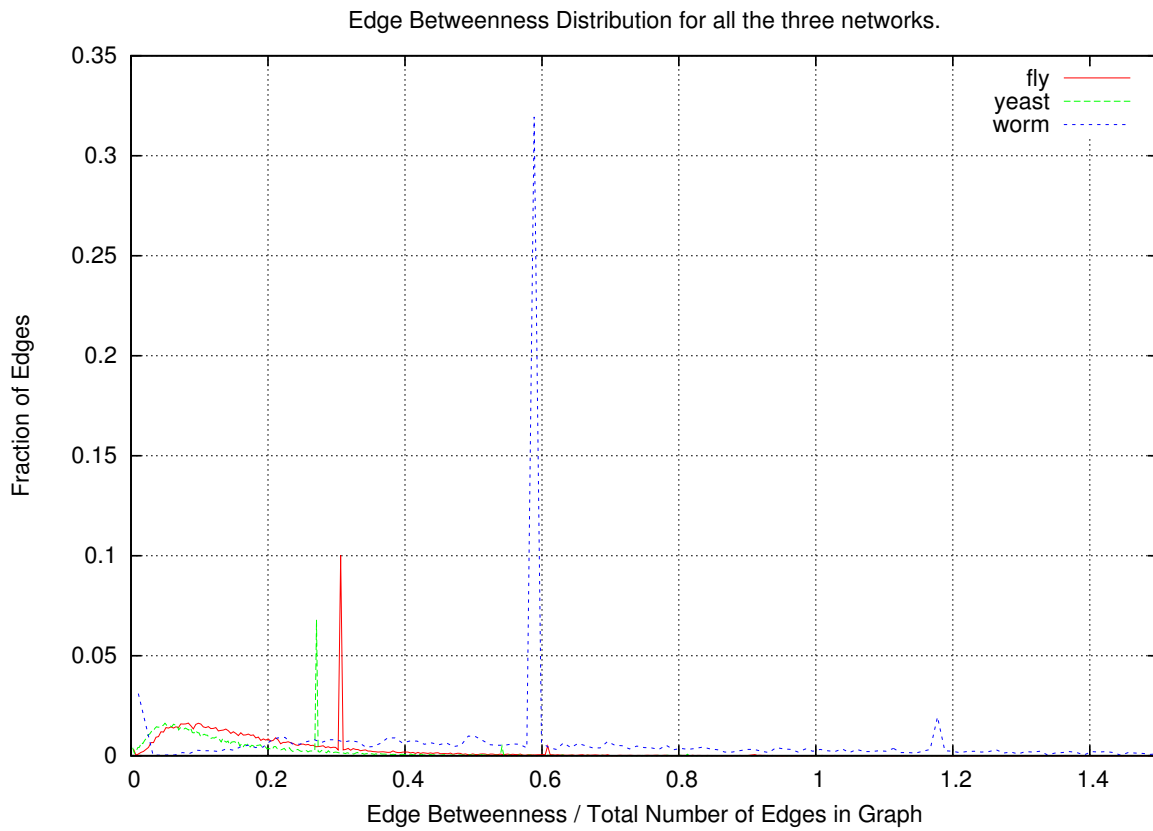


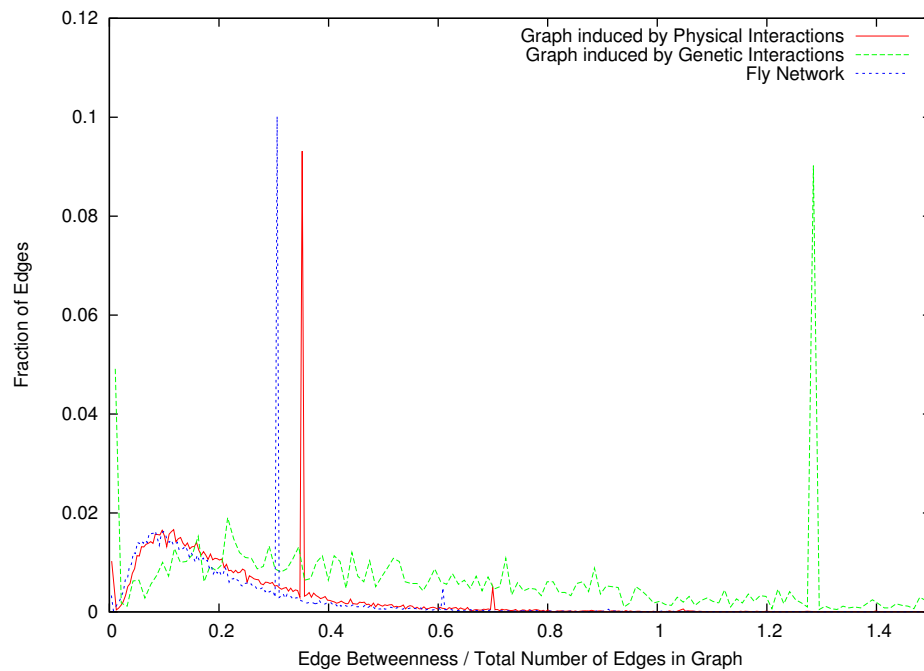
Figure 6.7: Edge betweenness distribution for all the three networks. We divide each edge betweenness value by the total number of edges in the graph.

by dividing by the total number of edges in the graph, the spikes occurs very close to each other, for the yeast and fly network (Figure 6.7). We had not anticipated that the edge betweenness distribution would have this shape. The rest of this chapter describes our attempts to explain why the edge betweenness distribution of biological networks has the observed properties.

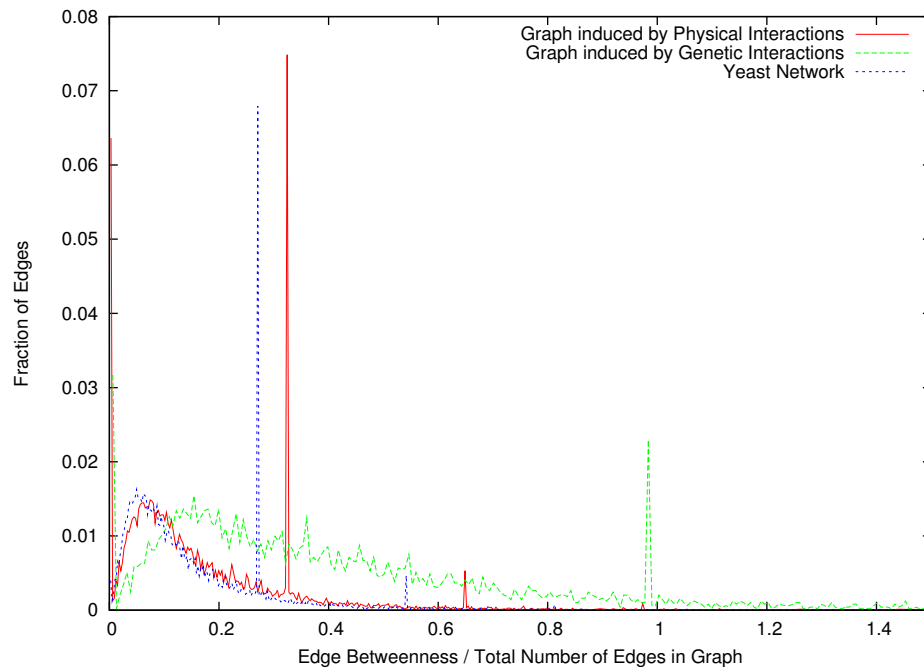
6.2.2 Edge Betweenness of Synthetically Lethal Interactions

We conjectured that the spike in the yeast and fly network may be caused by interactions in the graph which were synthetically lethal i.e., genetic interactions. Synthetically lethal interactions often occur between proteins participating in different pathways. Therefore, it is possible that these interactions acting as bridges in the physical network, leading them to have high edge betweenness values. Hence, we decided to compute the edge betweenness distribution for the graph induced by synthetically lethal interactions and the graph induced by the physical interactions separately, for the yeast and fly network. We also plotted the edge betweenness distribution for the graph induced by synthetically lethal interactions and the graph induced by the physical interactions using the edge betweenness values from the computation of edge betweenness for the original yeast and fly network.

From figure 6.8 and 6.9, it is clear that although there is a large number of synthetically lethal interactions in the spike, removal of those edges does not affect

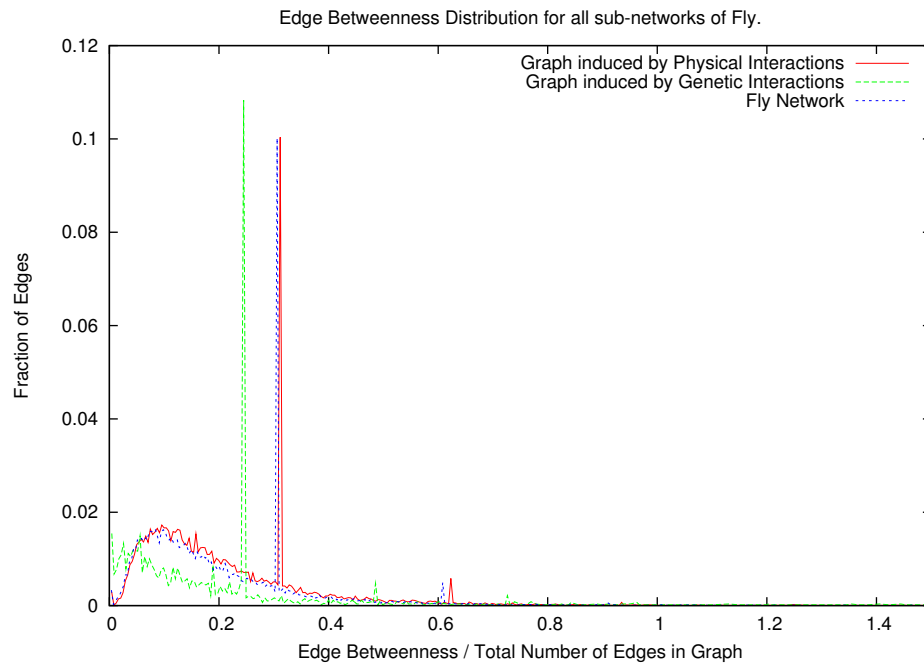


(a) Edge Betweenness Distribution of all sub-networks in Fly.

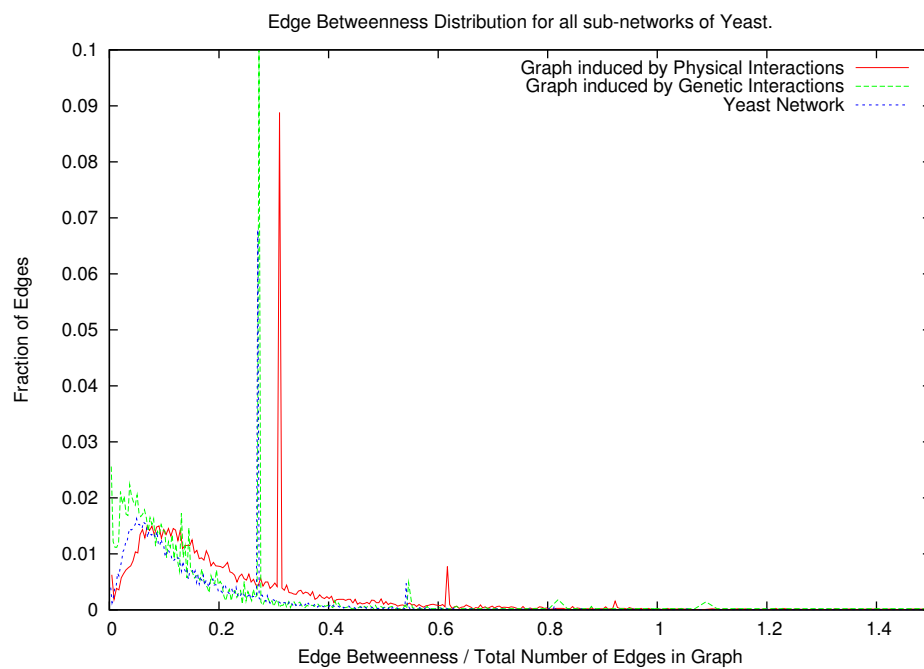


(b) Edge Betweenness Distribution of all sub-networks in Yeast.

Figure 6.8: (a) Edge betweenness distribution for all three networks in Fly considering the edge betweenness value from the original network. (b) Edge betweenness distribution for all three networks in Yeast considering the edge betweenness value from the original network.



(a) Edge Betweenness Distribution of all sub-networks in Fly.



(b) Edge Betweenness Distribution of all sub-networks in Yeast.

Figure 6.9: (a) Edge betweenness distribution for all three networks in Fly. (b) Edge betweenness distribution for all three networks in Yeast.

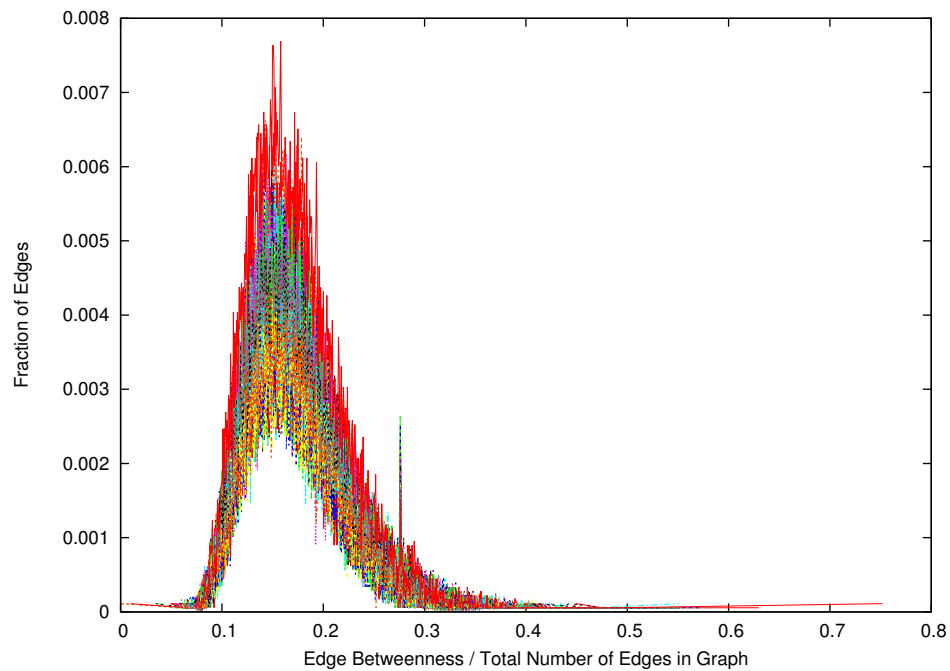
the shape of the distribution. In fact from figure 6.9, we can clearly see that after the removal of the synthetic lethal interaction edges from the graph, the edge betweenness distribution still has the spike, but the value of edge betweenness at which the spike occurs is greater. Thus our conjecture was incorrect.

6.3 Randomized Analysis

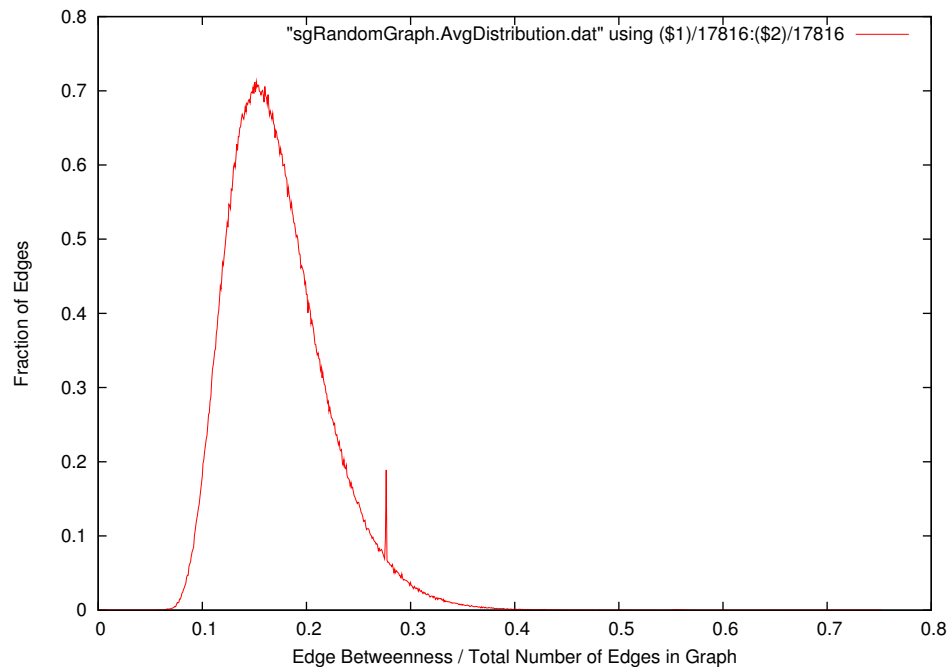
We decided to investigate whether the observed edge betweenness distribution were a property solely of the biological networks analysed or whether graph generation models (such as those described in Chapter 3) could yield graphs with similar edge betweenness distribution. We used the JUNG⁴⁵ framework to generate many types of random graphs. All the random graphs that we generated had the same number of vertices and edges as the yeast network i.e., 4920 vertices and 17816 edges.

6.3.1 Simple Random Graphs

The input to the simple random graph generator was the number of vertices $n = 4920$ and the number of edges $m = 17816$. The method first creates n vertices and then constructs the edges uniformly at random from the set of all edges. We generated a hundred simple random graphs and computed the edge betweenness values for all the edges in these graphs.



(a) Edge Betweenness Distribution of 100 Simple Random Graphs



(b) Average Edge Betweenness Distribution for 100 Simple Random Graphs

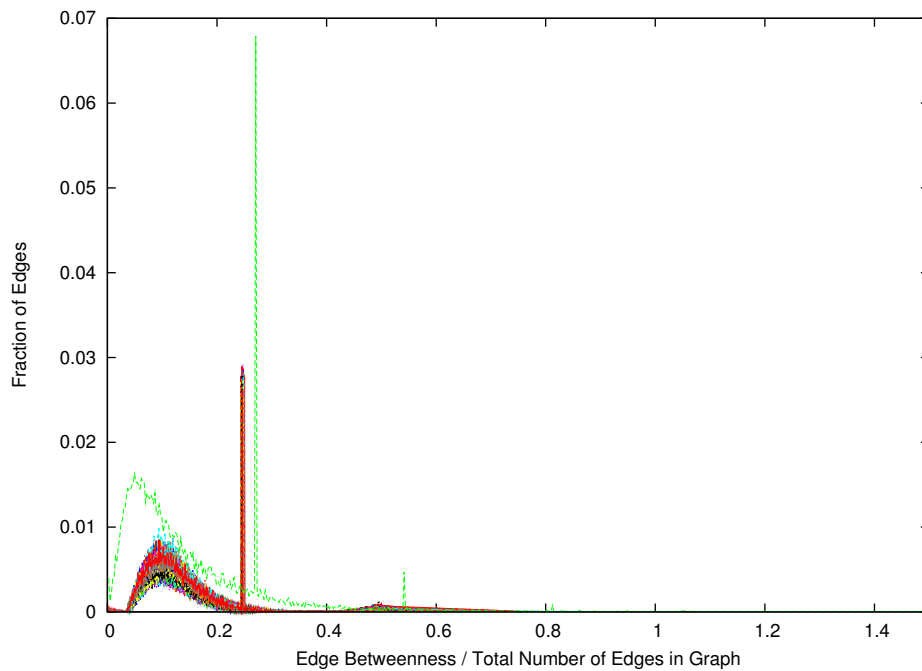
Figure 6.10: (a)Edge Betweenness Distribution of 100 Simple Random Graphs. We divide each edge betweenness value by total number of edges in the graph.(b)Average Edge Betweenness Distribution for 100 Simple Random Graphs.We divide each edge betweenness value by total number of edges in the graph

From figure 6.10, it is clear that the edge betweenness distribution of biological networks is very different from the edge betweenness distribution of simple random graphs.

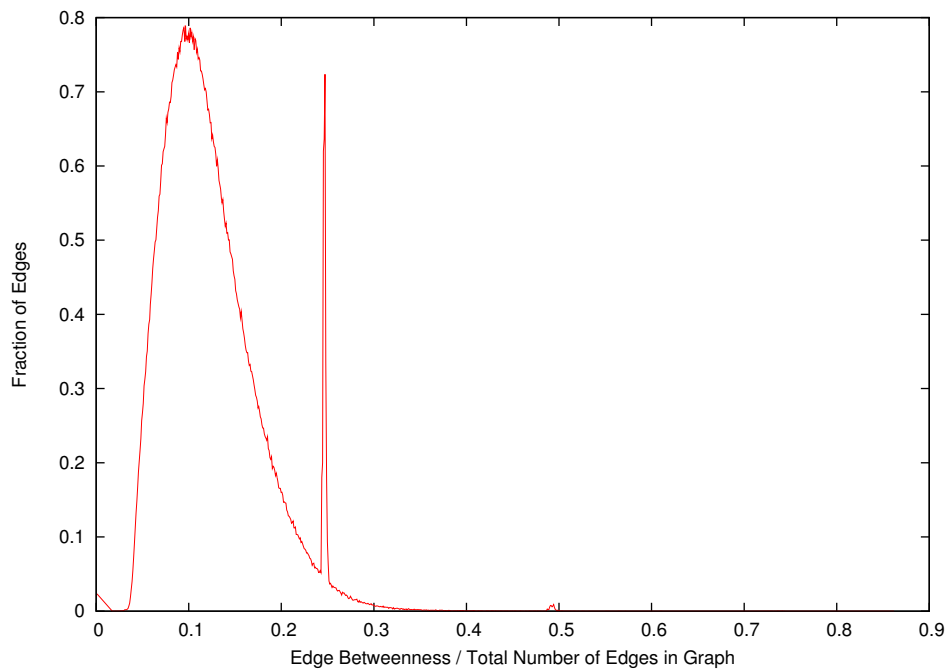
6.3.2 Eppstein Wang Power Law Random Graphs

In the Eppstein Wang²⁸ random graph generator, the input is the number of vertices n , the number of edges m and the model parameter r , which is the number of times the algorithm is run. The larger this parameter, the better the resulting graph's degree distribution approximates a power law. We generated a hundred random graphs with 4920 vertices, 17816 edges and with the value of r set to 10^7 . The edge betweenness values for all the edges were calculated, for all the hundred graphs.

In figure 6.11, we can see that there is a spike in the edge betweenness distribution, this spike was also noticed at in figure 6.6, edge betweenness distribution of the yeast network. On closer inspection, it can also be seen that the edge betweenness value at which the spikes occurs is quite close for both the figures. Although the value of edge betweenness is very close, the height of the spike is different in both figures.



(a) Edge Betweenness Distribution of 100 Eppstein Wang Power law Random Graphs



(b) Average Edge Betweenness Distribution for 100 Eppstein Wang Power law Random Graphs

Figure 6.11: (a) Edge Betweenness Distribution of 100 Eppstein Wang Power law Random Graphs. (b) Average Edge Betweenness Distribution for 100 Eppstein Wang Power law Random Graphs. We divide each edge betweenness value by total number of edges in the graph.

6.4 Random Graphs with Similar Degree

Distribution as the Biological Networks

In the previous section, we observed that the edge betweenness distribution of the yeast network and the edge betweenness distribution of the scale-free network generated by the Eppstein Wang model were similar. This observation motivated us to check whether the peculiar properties of the edge betweenness distribution that we had observed held true for any network with the same degree distribution as the biological networks. To this end, we constructed hundred random graphs each with the same degree distribution as the yeast, worm and fly interaction networks. The procedure we followed to construct the random graphs is as follows: We first created n nodes and assigned to each node a degree based on the degree distribution given. Next, for each node, we created a number of stubs equal to the degree of the node. Finally, we randomly paired stubs with each other and connected the two nodes corresponding to each pair of stubs as an edge. This process created self loops and multiple edges between the same pair of nodes. We deleted the self loops and kept only one copy of each multiple edge.

Remarkably, from figures 6.12, 6.13 and 6.14, it is clear that the random graphs generated with the same degree distribution also have an edge betweenness distribution very similar to the original network. In this case, the position of the spike for figures 6.12, 6.13 and 6.14, is at nearly the same value of edge

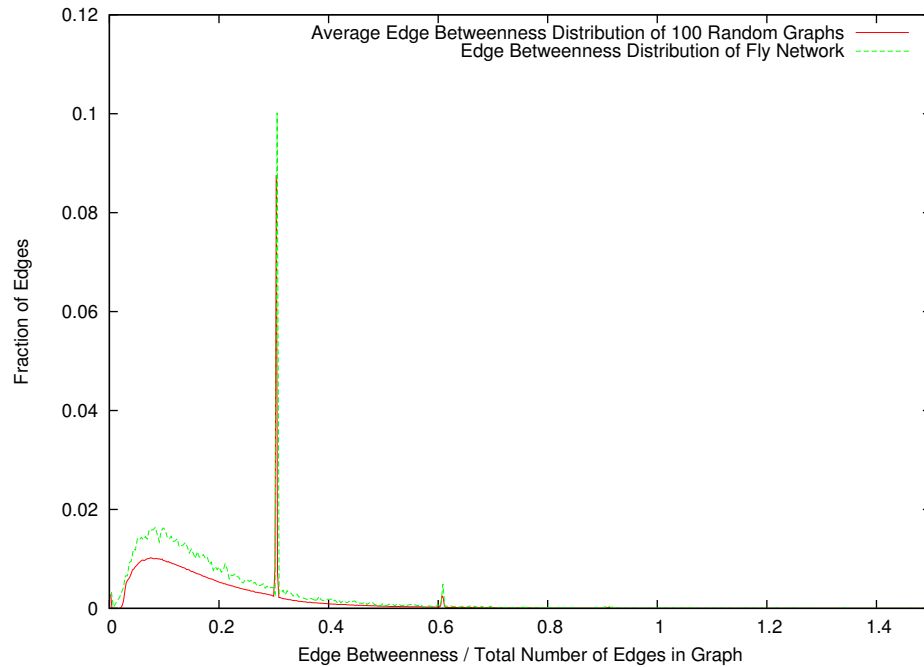


Figure 6.12: Edge betweenness distribution of the fly network and the average edge betweenness distribution of 100 random networks with the same degree distribution as the fly network.

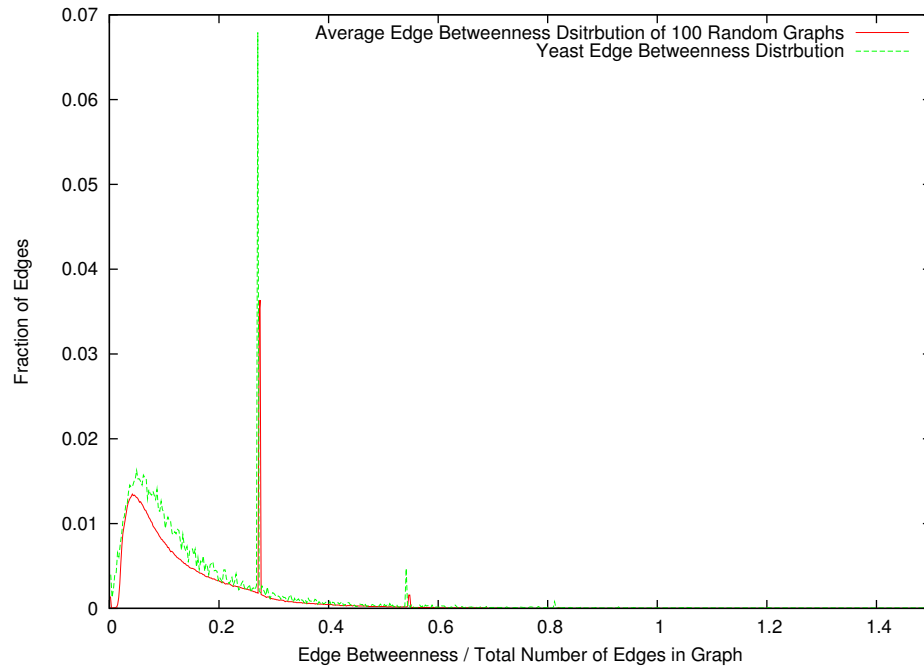


Figure 6.13: Edge betweenness distribution of the yeast network and the average edge betweenness distribution of 100 random networks with the same degree distribution as the yeast network.

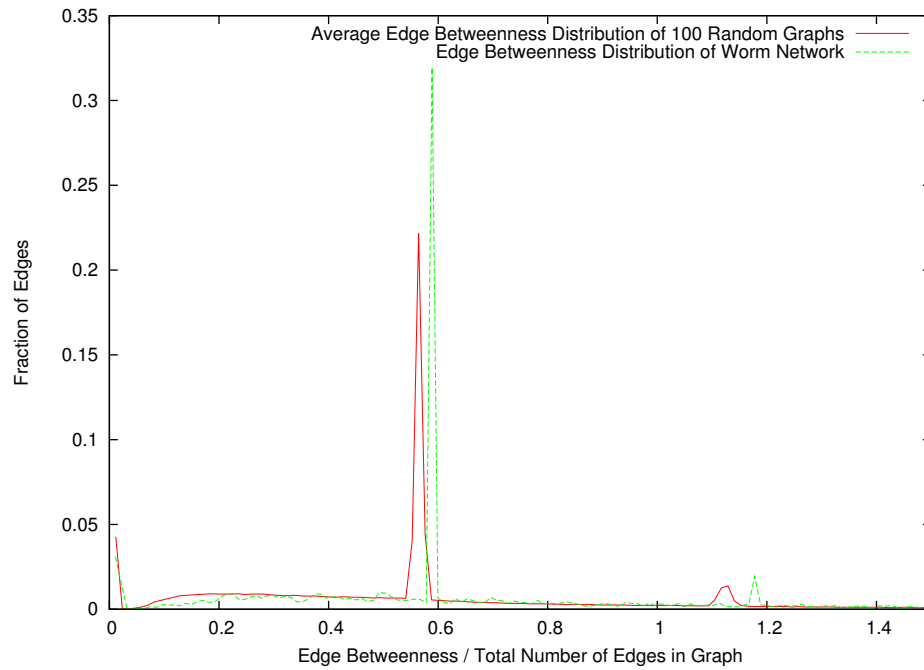


Figure 6.14: Edge betweenness distribution of the worm network and the average edge betweenness distribution of 100 random networks with the same degree distribution as the worm network.

betweenness at which the large spike occurs in the edge betweenness distribution of the original networks. Since the shape of the distribution is same for both the biological graphs and the random graphs, we have empirically demonstrated that the edge betweenness distribution that we are seeing is a property of the degree distribution of the graph, atleast when the degree distribution follows a power law.

6.5 Further Analysis on Random Graphs

To investigate this property further, we created graphs of different sizes and densities. The degree distribution of these graphs followed a power law with different values for the power law exponent. We wanted to check if the edge betweenness distribution of these graphs had a shape similar to the one we were observing. We wanted to know if there was a relation between the position and size of the spike to the power law exponent or the density of the graph.

The size of the graph is defined as the number of nodes present in the graph. The density of a graph is defined as the ratio of the number of edges m to the number of nodes n in the graph. We created the degree distribution of the graphs that followed a power law by setting the value of the size, the power law exponent and the density of the graphs. The procedure used to create the degree distribution is as follows: We first calculated the maximum possible degree of a node in the graph using the value of density m and power law exponent γ that are given, and the following relation:

$$m/n \geq \sum_{i=1}^{maxdegree} \frac{i^{1-\gamma}}{i^{-\gamma}} \quad (6.1)$$

We did not create the degree distribution, if the maximum degree that we calculated exceeded the size of the graph. Once we had the maximum possible degree of a node in the graph using the relation (6.1), we assigned the number of nodes k' with a certain degree k using the following relation:

$$k' = k^{-\gamma} \quad (6.2)$$

We created 124 degree distributions of graphs, with power law exponent ranging from 1 to 2.4 with increments of 0.2, with density ranging from 1 to 4.6 with increments of 0.4 and sizes, 1000 and 3000. Twenty random graphs were generated for each of the 124 degree distribution using the procedure we described in the previous section.

From figures 6.15, 6.16, 6.17, 6.18, 6.19 and 6.20, we observe that the edge betweenness distribution for graphs whose degree distribution follows power law do have a shape similar to the one we have seen earlier. We can see from figures 6.15, 6.16 and 6.17, that the position of the spike seems to be converging at a point as the density increases for all values of power law exponent, and as the density increases the position of the spike occurs at lower values of edge betweenness.

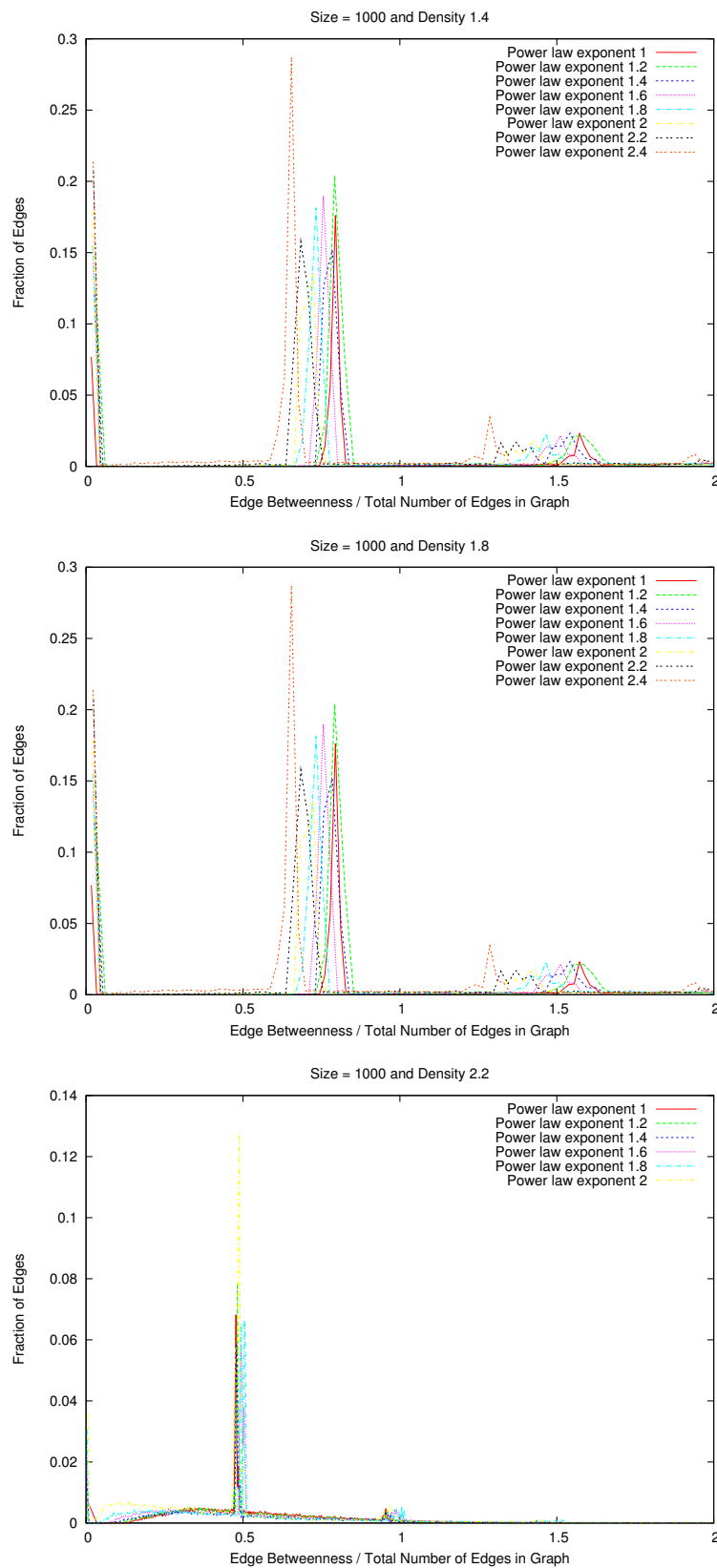


Figure 6.15: Each sub figure shows the average edge betweenness distribution for graphs with size 1000 and same density, but different values for the power law exponent.

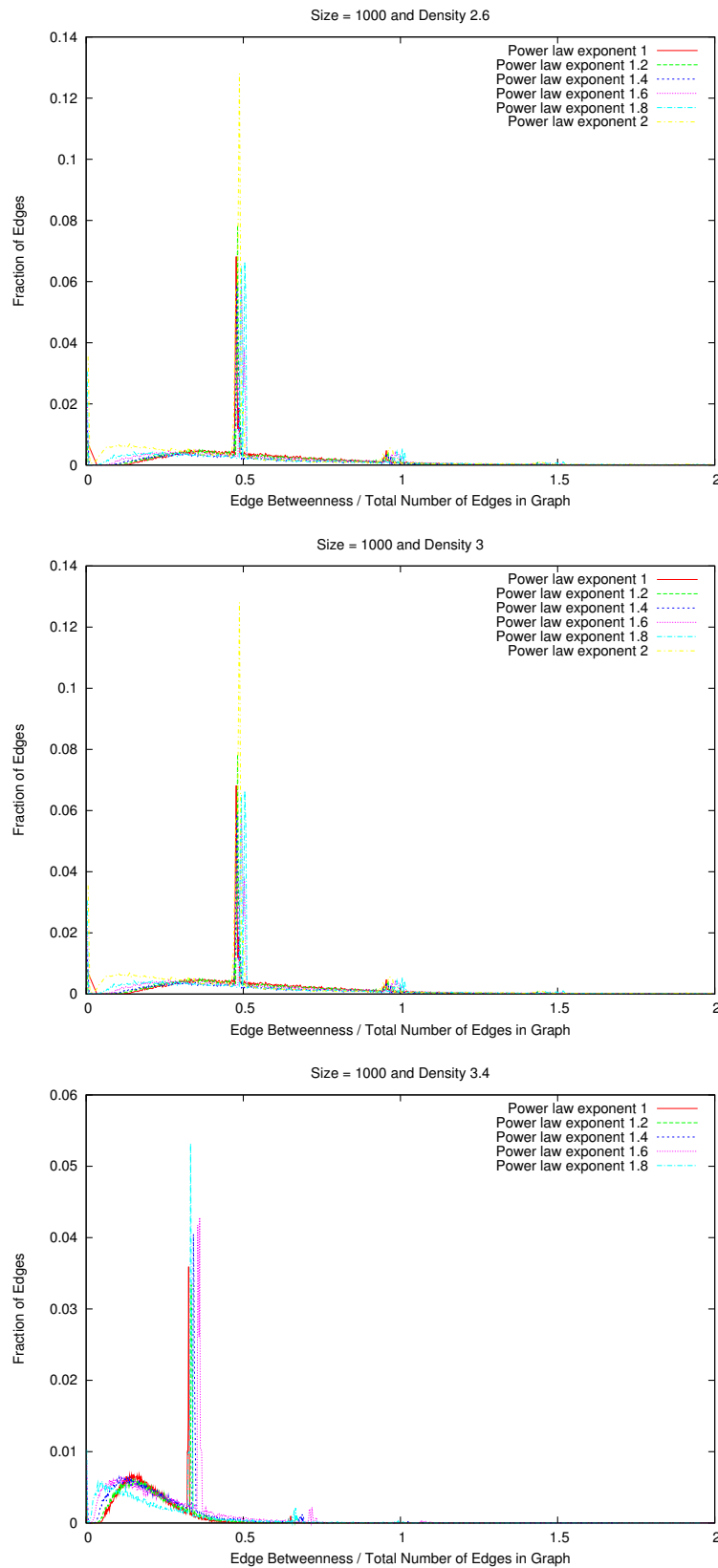


Figure 6.16: Each sub figure shows the average edge betweenness distribution for graphs with size 1000 and same density, but different values for the power law exponent.

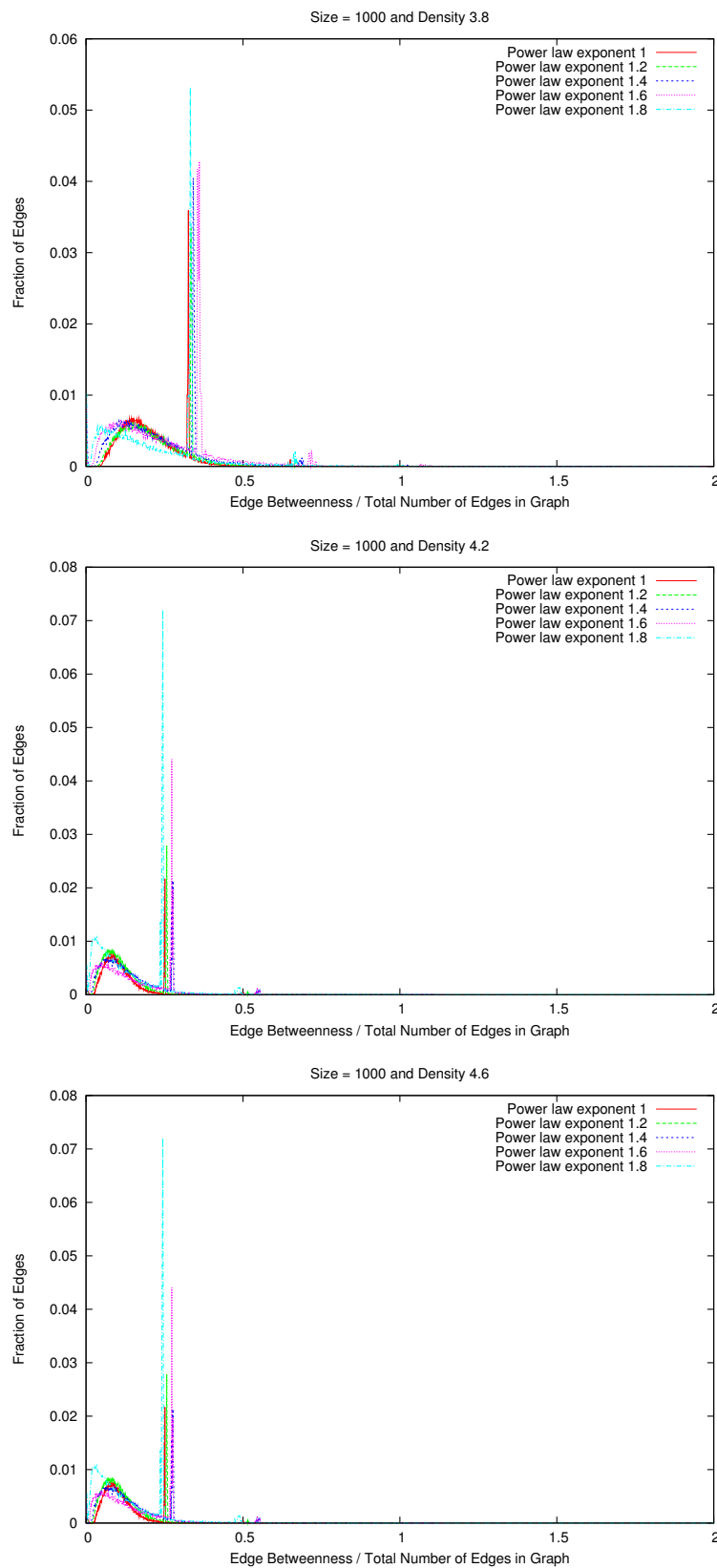


Figure 6.17: Each sub figure shows the average edge betweenness distribution for graphs with size 1000 and same density, but different values for the power law exponent.

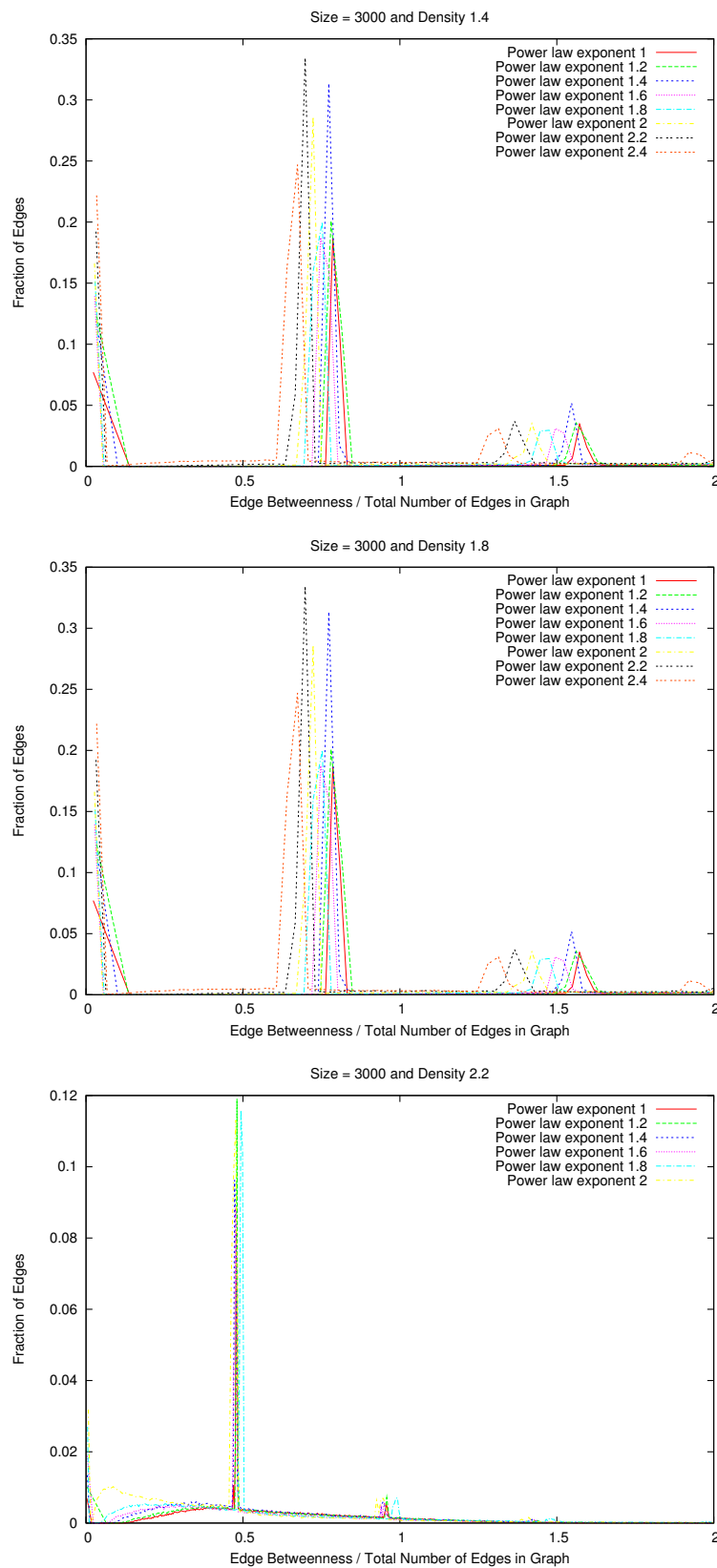


Figure 6.18: Each sub figure shows the average edge betweenness distribution for graphs with size 3000 and same density, but different values for the power law exponent.

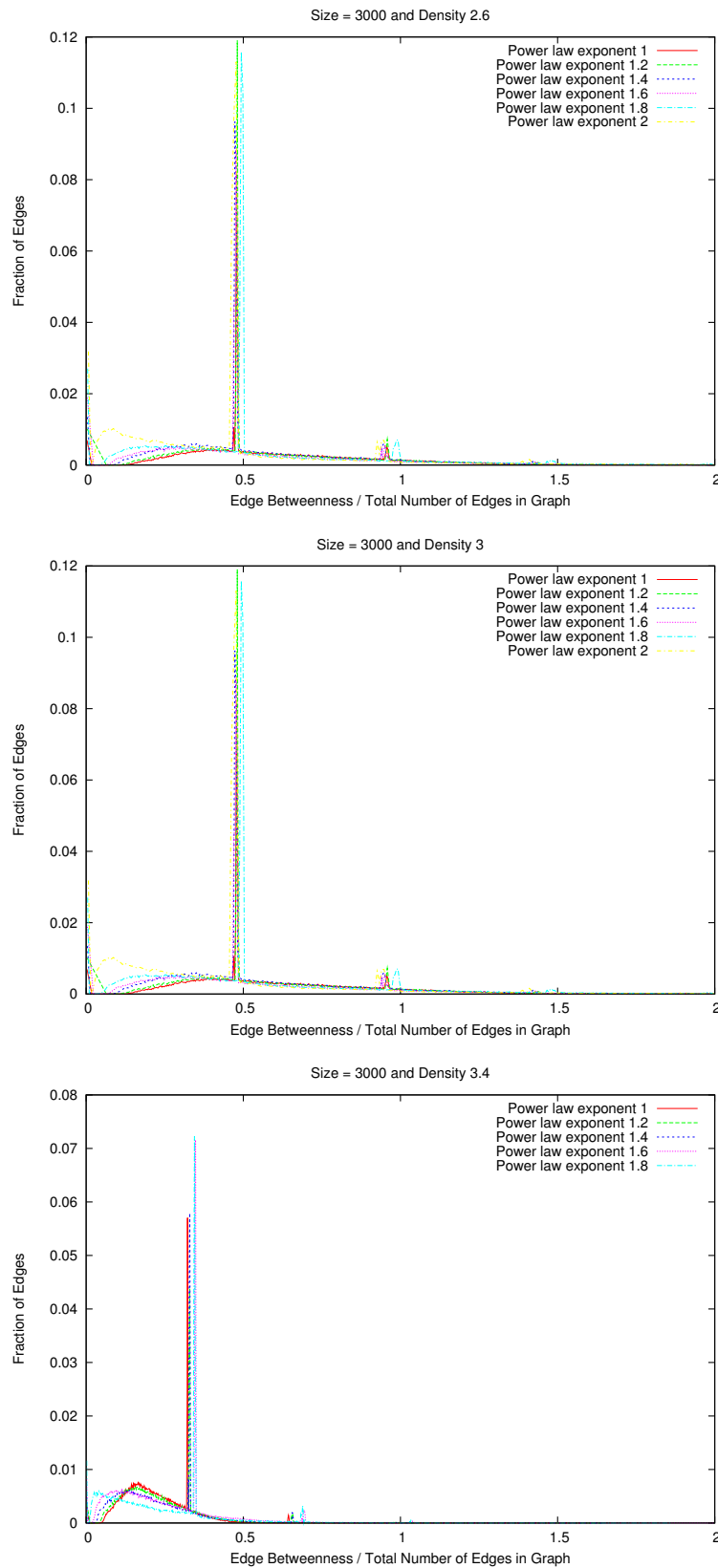


Figure 6.19: Each sub figure shows the average edge betweenness distribution for graphs with size 3000 and same density, but different values for the power law exponent.

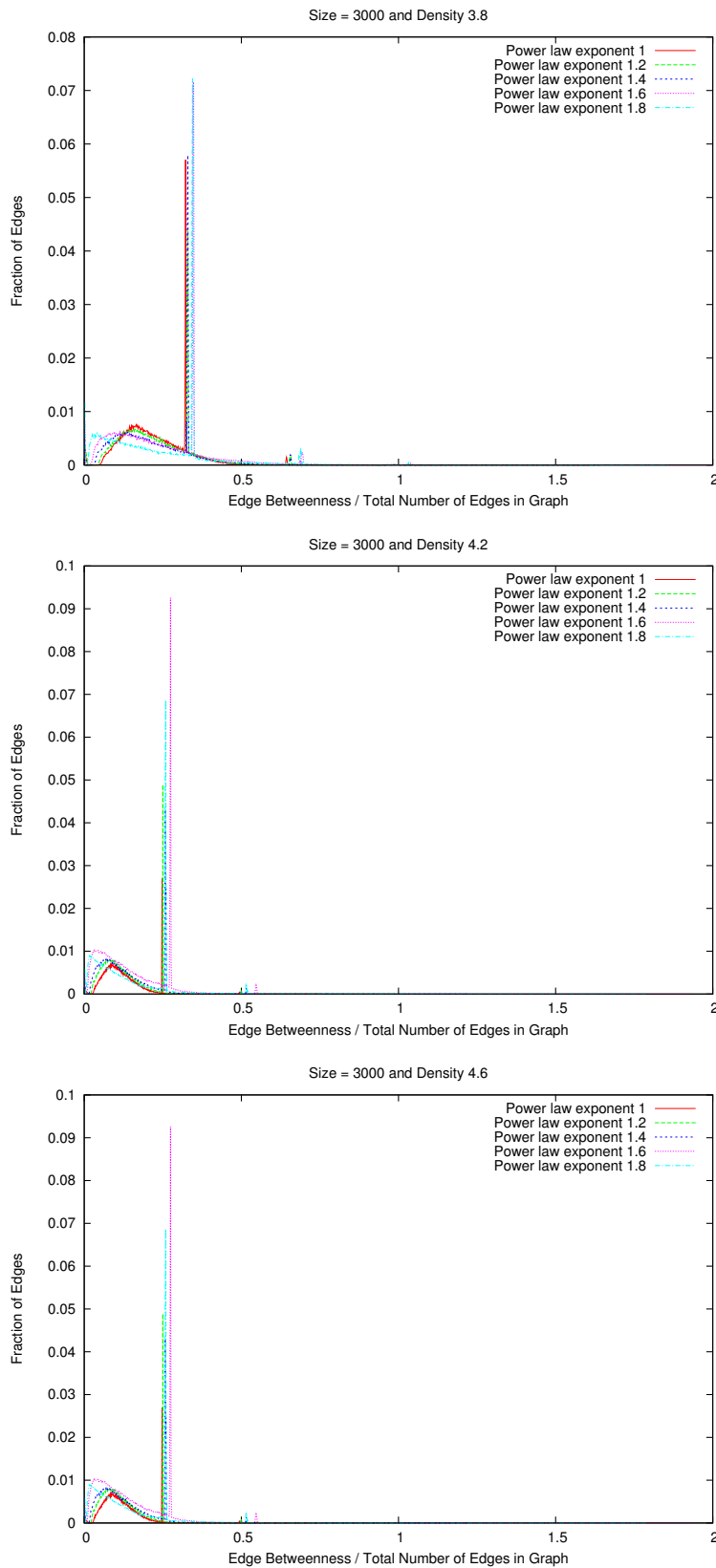


Figure 6.20: Each sub figure shows the average edge betweenness distribution for graphs with size 3000 and same density, but different values for the power law exponent.

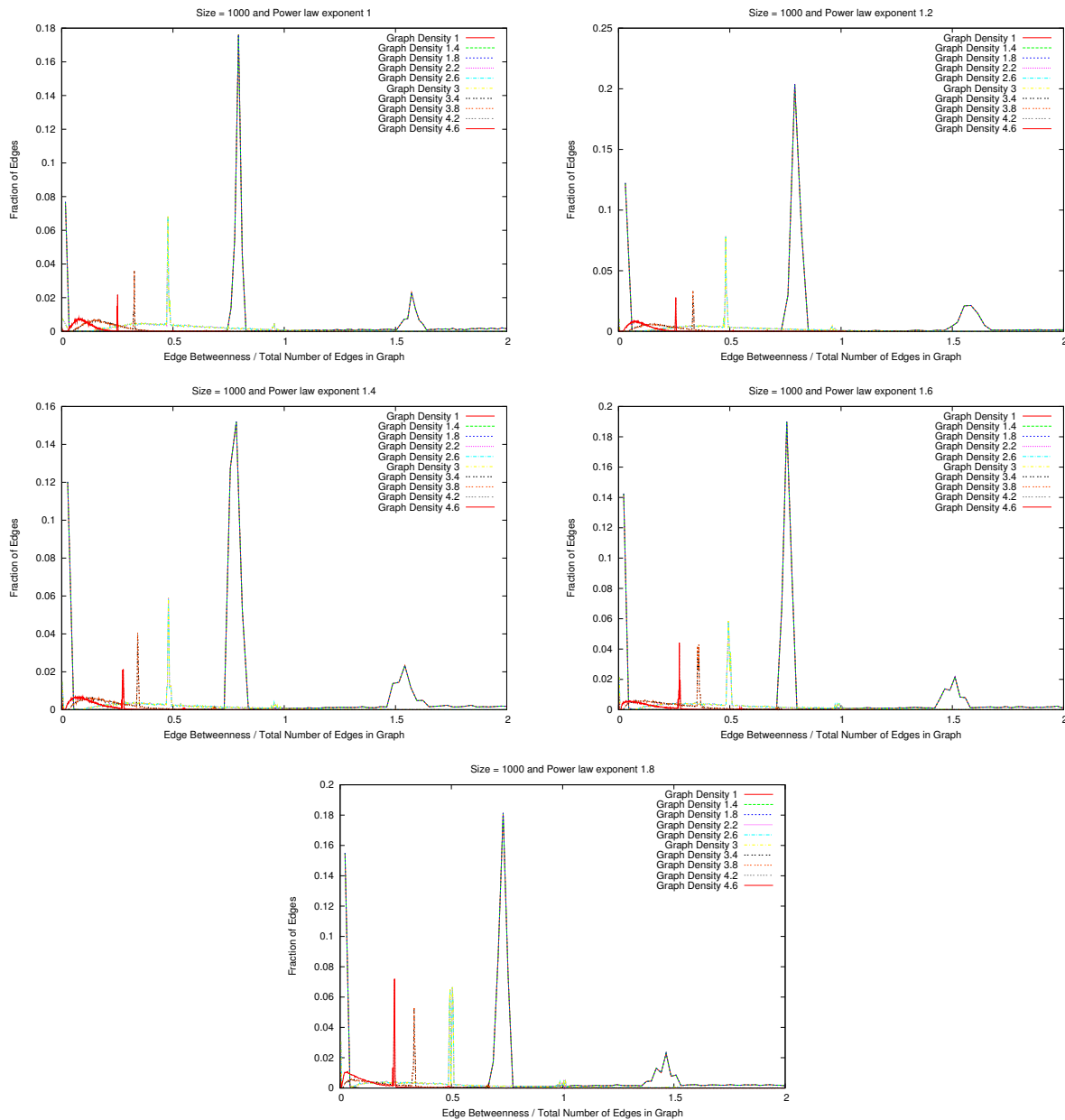


Figure 6.21: Each sub figure shows the average edge betweenness distribution for graphs with size 1000 and same power law exponent, but different values for the density.

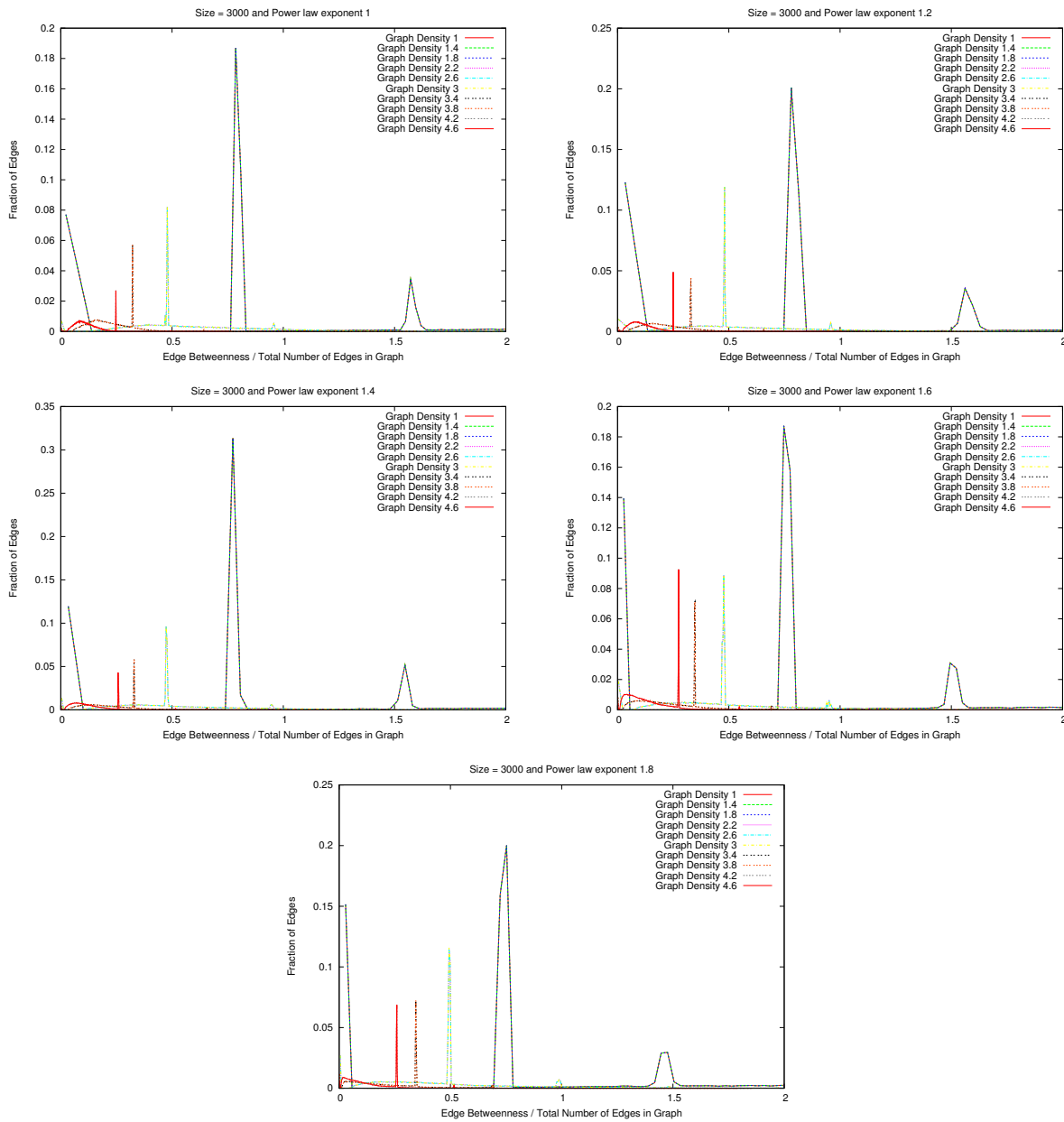


Figure 6.22: Each sub figure shows the average edge betweenness distribution for graphs with size 3000 and same power law exponent, but different values for the density.

From figures 6.21 and 6.22, we observe that the value at which the spike occurs remains nearly the same for graphs with different densities, even as the power law exponent is increasing.

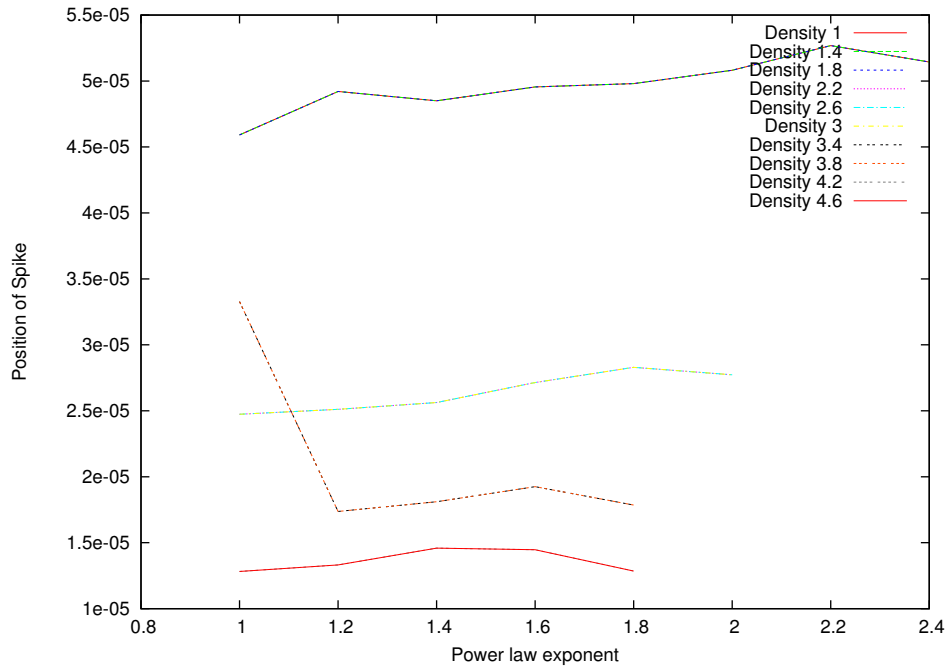


Figure 6.23: Power law exponent vs. Position of Spike, for different values of Density for graph with size 1000.

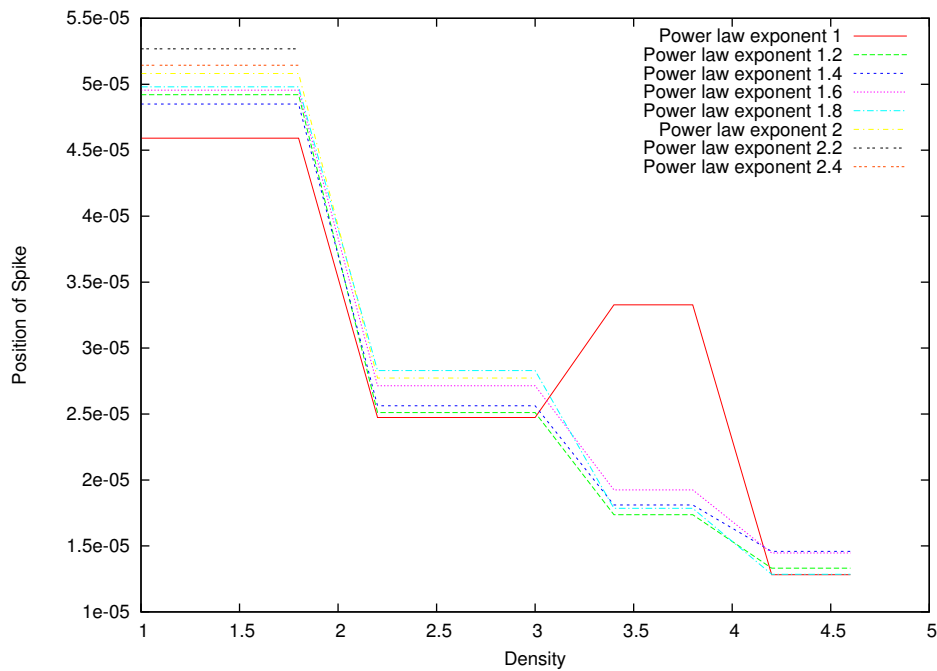


Figure 6.24: Density vs. Position of Spike, for different values of Power law exponent for graph with size 1000.

From figure 6.23, we observe that the value of edge betweenness, at which the spike occurs, remains nearly the same across all values of the power law exponent, for different densities of the graph. From figure 6.24, it is clear that value of edge betweenness, at which the spike occurs decreases as the density increases, for all values of the power law exponent.

Chapter 7

Conclusions

We applied the graph theoretic property of betweenness centrality on biological networks.

We first computed the vertex betweenness properties for all the biological networks.

We observed that the vertex betweenness distribution follows a power law for all the three networks, with exponents 4.4, 2.4 and 4.2. We also noted that vertex betweenness and vertex degree are highly correlated.

We saw some interesting properties in the edge betweenness distribution for all the biological networks. Each network has a large fraction of edges with nearly the same edge betweenness value.

We generated random graphs with the same degree distribution as the original biological networks. To our great surprise, we observed that the edge betweenness

distribution of all random graphs had the same shape as the edge betweenness distribution of the original biological networks. We also observed that the value at which the spike occurs in the average edge betweenness distribution of the random graphs is nearly the same value at which the spike occurs in the edge betweenness distribution of the original network.

We also generated random graphs with different sizes, densities and, whose degree distribution followed a power law, with different values of the power law exponent. The edge betweenness distribution of these graphs have also exhibited the shape we have been observing so far. We observed that the value of edge betweenness at which the spike occurred remained nearly the same for increasing power law exponent, for all values of densities of the graph. We also observed that the value of edge betweenness at which the spike occurred decreased for increasing density of the graph, for all values of the power law exponent.

From these analysis and results, we conjecture that graphs whose degree distribution follows a power law, will have an edge betweenness distribution with a large fraction of edges with nearly the same edge betweenness. We leave a formal proof of conjecture as an open problem.

Bibliography

1. Marcotte, E. M., Pellegrini, M., Thompson, M. J., Yeates, T. O., and Eisenberg, D. A combined algorithm for genome-wide prediction of protein function. *Nature* 402(6757):83–86, November, 1999.
2. Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., and Takagi, T. Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast* 18(6):523–531, April, 2001.
3. Vazquez, A., Flammini, A., Maritan, A., and Vespignani, A. Global protein function prediction from protein-protein interaction networks. *Nat Biotechnol* 21(6):697–700, June, 2003.
4. Karaoz, U., Murali, T. M., Letovsky, S., Zheng, Y., Ding, C., Cantor, C. R., and Kasif, S. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc Natl Acad Sci U S A* 101(9):2888–2893, March, 2004.

5. Zhou, X., Kao, M. C., and Wong, W. H. Transitive functional annotation by shortest-path analysis of gene expression data. *Proc Natl Acad Sci U S A* 99(20):12783–12788, October, 2002.
6. Barabasi, A.-L. and Albert, R. Emergence of scaling in random networks. *Science* 286(5439):509–512, October, 1999.
7. Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabasi, A. L. The large-scale organization of metabolic networks. *Nature* 407(6804):651–654, October, 2000.
8. Albert, R., Jeong, H., and Barabasi, A.-L. Error and attack tolerance of complex networks. *Nature* 406(6794):378–382, July, 2000.
9. Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* 393(6684):440–442, June, 1998.
10. Wuchty, S. and Stadler, P. F. Centers of complex networks. *J Theor Biol* 223(1):45–53, July, 2003.
11. Newman, M. E. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Phys Rev E Stat Nonlin Soft Matter Phys* 64(1 Pt 2), July, 2001.
12. Fields, S. and Song, O. A novel genetic system to detect protein-protein interactions. *Nature* 340(6230):245–246, July, 1989.

13. Uetz, P., Giot, L., Cagney, G., Mansfield, T. A., Judson, R. S., Knight, J. R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., Qureshi-Emili, A., Li, Y., Godwin, B., Conover, D., Kalbfleisch, T., Vijayadamodar, G., Yang, M., Johnston, M., Fields, S., and Rothberg, J. M. A comprehensive analysis of protein-protein interactions in *saccharomyces cerevisiae*. *Nature* 403(6770):623–627, February, 2000.
14. Ito, T., Tashiro, K., Muta, S., Ozawa, R., Chiba, T., Nishizawa, M., Yamamoto, K., Kuhara, S., and Sakaki, Y. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc Natl Acad Sci U S A* 97(3):1143–1147, February, 2000.
15. Giot, L., Bader, J. S., Brouwer, C., Chaudhuri, A., Kuang, B., Li, Y., Hao, Y. L., Ooi, C. E., Godwin, B., Vitols, E., Vijayadamodar, G., Pochart, P., Machineni, H., Welsh, M., Kong, Y., Zerhusen, B., Malcolm, R., Varrone, Z., Collis, A., Minto, M., Burgess, S., Mcdaniel, L., Stimpson, E., Spriggs, F., Williams, J., Neurath, K., Ioime, N., Agee, M., Voss, E., Furtak, K., Renzulli, R., Aanensen, N., Carroll, S., Bickelhaupt, E., Lazovatsky, Y., Dasilva, A., Zhong, J., Stanyon, C. A., Finley, R. L., White, K. P., Braverman, M., Jarvie, T., Gold, S., Leach, M., Knight, J., Shimkets, R. A., Mckenna, M. P., Chant, J., and Rothberg, J. M. A protein interaction map of *drosophila melanogaster*. *Science* 302(5651):1727–1736, December, 2003.

16. Li, S., Armstrong, C. M., Bertin, N., Ge, H., Milstein, S., Boxem, M., Vidalain, P. O., Han, J. D., Chesneau, A., Hao, T., Goldberg, D. S., Li, N., Martinez, M., Rual, J. F., Lamesch, P., Xu, L., Tewari, M., Wong, S. L., Zhang, L. V., Berriz, G. F., Jacotot, L., Vaglio, P., Reboul, J., Hirozane-Kishikawa, T., Li, Q., Gabel, H. W., Elewa, A., Baumgartner, B., Rose, D. J., Yu, H., Bosak, S., Sequerra, R., Fraser, A., Mango, S. E., Saxton, W. M., Strome, S., Van Den Heuvel, S., Piano, F., Vandenhaute, J., Sardet, C., Gerstein, M., Doucette-Stamm, L., Gunsalus, K. C., Harper, J. W., Cusick, M. E., Roth, F. P., Hill, D. E., and Vidal, M. A map of the interactome network of the metazoan *c. elegans*. *Science* 303(5657):540–543, January, 2004.
17. Gavin, A. C., Bsche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J. M., Michon, A. M., Cruciat, C. M., Remor, M., Hfert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., Rudi, T., Gnau, V., Bauch, A., Bastuck, S., Huhse, B., Leutwein, C., Heurtier, M. A., Copley, R. R., Edelmann, A., Querfurth, E., Rybin, V., Drewes, G., Raida, M., Bouwmeester, T., Bork, P., Seraphin, B., Kuster, B., Neubauer, G., and Superti-Furga, G. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415(6868):141–147, January, 2002.
18. Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., Moore, L., Adams, S. L., Millar, A., Taylor, P., Bennett, K., Boutilier, K., Yang, L., Wolting, C., Donaldson, I., Schandorff, S., Shewnarane, J., Vo, M., Taggart, J., Goudreault, M., Muskat, B.,

- Alfarano, C., Dewar, D., Lin, Z., Michalickova, K., Willems, A. R., Sassi, H., Nielsen, P. A., Rasmussen, K. J., Andersen, J. R., Johansen, L. E., Hansen, L. H., Jespersen, H., Podtelejnikov, A., Nielsen, E., Crawford, J., Poulsen, V., Srensen, B. D., Matthiesen, J., Hendrickson, R. C., Gleeson, F., Pawson, T., Moran, M. F., Durocher, D., Mann, M., Hogue, C. W., Figeys, D., and Tyers, M. Systematic identification of protein complexes in *saccharomyces cerevisiae* by mass spectrometry. *Nature* 415(6868):180–183, January, 2002.
19. Tong, A. H., Evangelista, M., Parsons, A. B., Xu, H., Bader, G. D., Pag, N., Robinson, M., Raghibizadeh, S., Hogue, C. W., Bussey, H., Andrews, B., Tyers, M., and Boone, C. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science* 294(5550):2364–2368, December, 2001.
20. Gilbert, D. Biomolecular interaction network database. *Brief Bioinform* 6(2):194–198, June, 2005.
21. Xenarios, I., Rice, D. W., Salwinski, L., Baron, M. K., Marcotte, E. M., and Eisenberg, D. Dip: the database of interacting proteins. *Nucleic Acids Res* 28(1):289–291, January, 2000.
22. Breitkreutz, B. J., Stark, C., and Tyers, M. The grid: the general repository for interaction datasets. *Genome Biol* 4(3), 2003.
23. Goldberg, D. S. and Roth, F. P. Assessing experimentally derived interactions in a small world. *Proc Natl Acad Sci U S A* 100(8):4372–4376, April, 2003.

24. Wagner, A. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol Biol Evol* 18(7):1283–1292, July, 2001.
25. Wuchty, S., Oltvai, Z. N., and Barabasi, A. L. Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nat Genet* 35(2):176–179, October, 2003.
26. Erdos, P. and Renyi, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* 5:17–61, 1960.
27. Barabasi, A. L. and Albert, R. Emergence of scaling in random networks. *Science* 286(5439):509–512, October, 1999.
28. Eppstein, D. and Wang, J. A steady state model for graph power laws, , Mar, 2002.
29. Freeman, L. A set of measures of centrality based on betweenness. *Sociometry* 40:35–41, 1977.
30. Freeman, L. Centrality in social networks:conceptual clarification. *Social Networks* 1:215–239, 1979.
31. Klovdahl, A. S., Graviss, E. A., Yaganehdoost, A., Ross, M. W., Wanger, A., Adams, G. J., and Musser, J. M. Networks and tuberculosis: an undetected community outbreak involving public places. *Soc Sci Med* 52(5):681–694, March, 2001.

32. Goh, K. I., Kahng, B., and Kim, D. Universal behavior of load distribution in scale-free networks. *Phys Rev Lett* 87(27 Pt 1), December, 2001.
33. Goh, K. I., Oh, E., Jeong, H., Kahng, B., and Kim, D. Classification of scale-free networks. *Proc Natl Acad Sci U S A* 99(20):12583–12588, October, 2002.
34. Kim, D. H., Noh, J. D., and Jeong, H. Scale-free trees: the skeletons of complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 70(4 Pt 2), October, 2004.
35. Holme, P. and Kim, B. J. Vertex overload breakdown in evolving networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 65(6 Pt 2), June, 2002.
36. Holme, P. Edge overload breakdown in evolving networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 66(3 Pt 2A), September, 2002.
37. Holme, P., Kim, B. J., Yoon, C. N., and Han, S. K. Attack vulnerability of complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 65(5 Pt 2), May, 2002.
38. Girvan, M. and Newman, M. E. Community structure in social and biological networks. *Proc Natl Acad Sci U S A* 99(12):7821–7826, June, 2002.
39. Holme, P., Huss, M., and Jeong, H. Subnetwork hierarchies of biochemical pathways. *Bioinformatics* 19(4):532–538, March, 2003.
40. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. Defining and identifying communities in networks. *Proc Natl Acad Sci U S A* 101(9):2658–2663, March, 2004.

41. Wilkinson, D. M. and Huberman, B. A. A method for finding communities of related genes. *Proc Natl Acad Sci U S A* 101 Suppl 1:5241–5248, April, 2004.
42. Dunn, R., Dudbridge, F., and Sanderson, C. M. The use of edge-betweenness clustering to investigate biological function in protein interaction networks. *BMC Bioinformatics* 6(1), March, 2005.
43. Joy, M. P., Brock, A., Ingber, D. E., and Huang, S. High-betweenness proteins in the yeast protein interaction network. *J Biomed Biotechnol* 2005(2):96–103, 2005.
44. Brandes, U. A faster algorithm for betweenness centrality, , 2001.
45. Jung:java universal network/graph framework. <http://jung.sourceforge.net>, 2005.

Vita

Shivaram Narayanan graduated with a Bachelor's degree in Computer Science from Indian Institute of Technology, Kharagpur, India in May 2003. Since August 2003 he has been a Master's student in the Department of Computer Science at Virginia Polytechnic Institute and State University. After graduating in December 2005, he will join the Security Infrastructure Team in S.W.I.F.T as a software developer.