Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Sijin Cherupilly Abdulkarim

Entitled
GRAPH BASED MINING ON WEIGHTED DIRECTED GRAPHS FOR SUBNETWORKS AND PATH DISCOVERY

For the degree of Master of Science

Is approved by the final examining committee:

Dr. Mathew J Palakal
Chair

Dr. Shiaofen Fang

Dr. Yuni Xia

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Mathew J Palakal

Approved by: Shiaofen Fang 04/11/2011
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

GRAPH BASED MINING ON WEIGHTED DIRECTED GRAPHS FOR SUBNETWORKS AND PATH DISCOVERY

For the degree of Master of Science

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Sijin Cherupilly Abdulkarim

Printed Name and Signature of Candidate

04/12/2011

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

GRAPH BASED MINING ON WEIGHTED DIRECTED GRAPHS FOR SUBNETWORKS AND PATH DISCOVERY

A Thesis

Submitted to the Faculty

of

Purdue University

by

Sijin Cherupilly Abdulkarim

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2011

Purdue University

Indianapolis, Indiana

ACKNOWLEDGEMENTS

I would like to take the opportunity to acknowledge some of the people who made my graduate study a memorable experience and made this thesis possible. Foremost, it is my sincere pleasure to express my deep and sincere gratitude to my advisor, Dr. Mathew J Palakal, for his guidance, motivation, feedback, encouragement, support, and patience during the course of my thesis. His input and efforts have been of great value to me.

I would like to thank the other members of my thesis committee, Dr. Shiaofen Fang and Dr. Yuni Xia for accepting my request to be a part of thesis committee. I must appreciate their efforts to review my work. I owe my sincere thanks to Indiana University for providing the financial support throughout my Master's program. This work was funded in part by a grant from the Department of Defense as part of the Cancer Care Engineering Project. I also want to thank Dr. Meeta Pradhan and members of the TiMAP Laboratory for their valuable suggestions during the course of this project.

Without the adequate academic preparation, my studies could not have been a successful experience. Hence, I would like to add my thanks to faculty and staff in the Department of Computer and Information science for their support in the course work.

I owe my loving thanks to my parents, and sisters for their encouragement and understanding. My loving thanks to Isaac Abraham for his help in my thesis writing and presentation. I would like to thank Gokul, Aditi, Kulin, Chetan, Tulip, Christina, Deepthi

for the help in proof reading. I would also like to thank my friends Sarang, Ruchin, Yahia, Madhura, Shashank and Deepika for their support and all the fun we have had in the last two years. On Top of all, I thank God for all his blessings and care.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Abdulkarim, Sijin Cherupilly. M.S., Purdue University, May 2011. Graph Based Mining on weighted directed graphs for subnetworks and path discovery. Major Professor: Mathew J Palakal.

Subnetwork or path mining is an emerging data mining problem in many areas including scientific and commercial applications. Graph modeling is one of the effective ways in representing real world networks. Many natural and man-made systems are structured in the form of networks. Traditional machine learning and data mining approaches assume data as a collection of homogenous objects that are independent of each other whereas network data are potentially heterogeneous and interlinked. In this paper we propose a novel algorithm to find subnetworks and Maximal paths from a weighted, directed network represented as a graph. The main objective of this study is to find meaningful Maximal paths from a given network based on three key parameters: node weight, edge weight, and direction. This algorithm is an effective way to extract Maximal paths from a network modeled based on a user's interest. Also, the proposed algorithm allows the user to incorporate weights to the nodes and edges of a biological network. The performance of the proposed technique was tested using a Colorectal Cancer biological network. The subnetworks and paths obtained through our network mining algorithm from the biological network were scored based on their biological

significance. The subnetworks and Maximal paths derived were verified using Metacore^{TM} as well as literature. The algorithm is developed into a tool where the user can input the node list and the edge list. The tool can also find out the upstream and downstream of a given entity (genes/proteins etc.) from the derived Maximal paths. The complexity of finding the algorithm is found to be $O(nlogn)$ in the best case and $O(n^2logn)$ in the worst case.

CHAPTER ONE: INTRODUCTION

1.1 Networks

A network is a collection of points, called vertices, and a collection of lines, called arcs, connecting these points. A graph is defined as a set of vertices V and set of edges E or pair of vertices. Edges are sometimes directed/ordered pairs and sometimes with weights. Systems taking the form of networks are also called “graphs” in much of the mathematical literature abound in the world. Examples include the Internet, the World Wide Web, social networks of acquaintance or other connections between individuals, organizational networks and networks of business relations between companies, neural networks, metabolic networks, food webs, distribution networks such as blood vessels or postal delivery routes, networks of citations between papers and many others. The study of networks, in the form of mathematical graph theory, is one of the fundamental pillars of discrete mathematics. Euler’s celebrated 1735 solution of the Konigsberg bridge problem is often cited as the first true proof in the theory of networks, and during the twentieth century, graph theory has developed into a substantial body of knowledge.

Recent years however have witnessed a substantial new movement, in network research with the focus shifting away from the analysis of single small graph and the properties of individual vertices or edges within such graphs to the consideration of large-scale statistical properties of graphs. This new approach has been driven largely by the

availability of computers and communication networks that allow us to gather and analyze data on a scale larger than what was possible previously.

1.1.1 Types of networks

A set of vertices joined by edges is only the simplest type of network; there are many ways in which a network can be complex than this network. For instance, there may be more than one different type of vertices in a network, or more than one different type of edges. Vertices or edges may have a variety of properties or numerical associated with them. Taking the example of social network of people, the vertices may represent sexes/genders, people of different nationalities, locations, ages, incomes, or many other things. Edges may represent friendship, but they could also represent animosity or professional acquaintance or geographical proximity. They can carry weights saying how well two people know each other. They can also be directed, pointing in only one direction. Graphs composed of directed edges are themselves called directed edges or arcs. A graph representing telephone calls, emails or messages between individuals would be directed, since each message goes in only one direction. Directed graphs can be cyclic, which means they contain closed loop of edges.

1.2 Networks in real world

In this section we look at what is known about the structure of networks of different types. Recent work on the mathematics of networks had been driven largely by observations of the properties of actual networks and attempts to model them, so network

data is the obvious starting point for a review such as this. In this paper we review four categories of networks.

1.2.1 Social network

A social network is a set of people or groups of people with some pattern of contact or interaction between them [129,154]. The patterns of friendships between individuals [100,121], business relationships between companies [93,121], and intermarriages between families [111] are all examples of networks that have been studied in the past. Some more examples of networks of this type would include a network of company directors, network of coauthorship among academics, in which individuals are linked if they have coauthored one or more papers, and coappearance networks in which individuals are linked by mention in the same context, particularly on Web pages or in newspaper articles. A personal connection between people where each edge between two people represents a letter or package sent by mail from one to another also falls in this category of networks. Another example can be a network of telephone calls where the vertices represent telephone numbers and the directed edges represents calls from one number to another.

1.2.2 Information networks

The second network category is called information networks. The classic example of an information network is the network of citation between academic papers [44]. Most learned articles cite previous work by others on related topics. These citations form a network in which the vertices are articles and a directed edge from article A to article B

indicates that A cites B. The structure of the citation network then reflects the structure of the information stored at its vertices, hence the term information network. Citation networks are acyclic because papers can only cite other papers that have already been published, not those that are yet to be written. Another very important example of an information network is World Wide Web which is linked together by hyperlinks from one page to another [19]. The web should not be with the internet, which is a physical network of computers linked together by optical fiber and other data connections. Unlike a citation network, the World Wide Web is cyclic.

1.2.3 Technological networks

The third class of networks is technological networks, man-made networks designed typically for distribution of some commodity or resource, such as electricity or information. The electric power is a good example. This is a network of high –voltage, three-phase transmission lines that span a country or a portion of a country. The telephone network and delivery networks such as those used by the post-office or parcel delivery companies also fall into this general category. Another very widely studied technological network is the internet, i.e. the network of physical connections between computers.

1.2.4 Biological networks

A number of biological systems can be usefully represented as networks. A classic example of a biological network is the network of metabolic pathways, which is a representation of metabolic substrates and products with directed edges joining them if a known metabolic reaction exists that acts on a given substrate and produces a given

product. Another network is the network of mechanistic physical interactions between proteins, which is usually referred to as a protein interaction network. An important class of biological network is the genetic regulatory network. The expression of a gene, i.e. the production by transcription and translation of the protein for which the gene codes, can be controlled by the presence of other proteins, both activators and inhibitors, so that the genome itself forms a switching network with vertices representing the proteins and directed edges representing dependence of protein production on the proteins at other vertices. Genetic regulatory networks were in fact the one of the first networked dynamical systems for which large-scale modeling attempts were made.

A well-known example of a biological network is the food web, in which the vertices represent species in an ecosystem and a directed edge from species X to species Y indicates that X preys on Y. Neural network are another class of biological networks of considerable importance. Blood vessels and the equivalent vascular networks in plants form the foundation for one of the most successful theoretical models of the effects of network structure on the behavior of a networked system.

1.3 Network mining verses data mining

Many natural and man-made systems are structured in the form of networks. Traditional machine learning and data mining approaches assume data as a collection of homogenous objects that are independent of each other whereas network data is potentially heterogeneous and interlinked. Objectives of network mining include entity identification, link prediction, link type prediction, discovery of communities of interest, discovery of infrequent or unusual patterns and link based object classifications.

Network mining can also be defined as data mining of data available within a network environment. Network data mining is concerned with discovering relationships and patterns in linked data, i.e. the interdependencies between data items at the lowest elemental level. These patterns can be revealing in and of themselves, whereas statistically summarized data patterns are informative in different but complementary ways.

1.4 Graph based mining

The need for mining structured data has increased in the past few years. One of the best studied data structures in computer science and discrete mathematics are graphs. Hence, it is no surprise that graph based data mining has become quite popular in the last few years. One of the most common ways to describing structural data is a graph representation. The graph is an abstract data structure consisting of vertices and edges which are relationship between vertices. Graph based data mining denotes a collection of algorithms for mining the relational aspects of data represented as a graph.

The huge amount of data available makes the desire for data mining grow. More and larger databases need to be searched to find interesting (and frequent) elements and relationships between them. Most often the data of interest is very complex. It is interesting to model complex data with the help of graphs consisting of nodes and edges that are often labeled to store additional information. Data can be best represented as networks which contain nodes and edges. Nodes represent objects and edges represent links between the objects. There may be different kinds of objects and different kinds of links in one network. Both object and link can be described by set of attributes (such as

weights). Having a graph database, it is always interesting to find common graphs in it, connections between different graphs, the subgraphs, the pathways and most importantly ranking the pathways.

Graph based mining for frequent patterns have recently developed into an area of intensive research. Recently, there aroused a large number of graphs with massive sizes and complex structures in many new applications, such as biological networks, social networks, and the web, demanding powerful data mining methods. Currently, a very popular area where graph based data mining is applied, is in drug discovery and compound synthesis. Most of the existing frequent subgraph mining algorithms are used to deal with undirected unweighted graphs. Consideration of weights and direction to these networks is a highly complex analysis. But in the real world, a lot of connections have directions, so directed weighted graph mining is more meaningful.

1.4.1 Application on social network

Traditional methods of machine learning and data mining, taking, as input, a random sample of homogenous objects from a single relation, may not be appropriate in social networks. The data comprising of social networks tend to be heterogeneous, multi relational, and semi-structure. As a result, a new field of research had emerged called network mining/graph based mining. The various outcomes on social network mining includes link-based object classification, object type prediction, kink type prediction, predicting link existence, link cardinality estimation, object reconciliation, group detection, sub graph detection, pattern discovery etc. In recent years the problem of how to find frequent subgraphs from a graph database has gained intensified and growing

attention. The first published algorithm in this area ‘Subdue’ that appeared in the mid-1990s, is the oldest algorithm, and yet is still used in various applications. In 1994, Agarwal and Srikant [1] introduced the concept of mining frequent patterns from a graph database. Recently, this method has been applied to large graph datasets in order to find the most common patterns from a large graph database. However, several questions remain unanswered, and there still remain unsolved problems for further investigation. Most of the current work is based on undirected graphs where the end point of each edge is from the same set of nodes. Another topic of interest in this field is weighted graphs, since the relationship between the edges/nodes provides extra information for data mining. Also, less attention has been paid to bipartite graphs.

1.4.2 Application on biological networks

Large-scale biological networks are often generated through various omics studies such as genomics, proteomics, bibliomics, and so on. In most cases, these are gene interaction or protein interaction networks. The nodes and edges of these networks along with the directionality of the edges have biological significance. By considering these networks as a database of graphs, it is always interesting to find the common graphs, connections between different graphs, the subgraphs, the Maximal paths or sub-paths, and most importantly, the ranking of derived sub graph structures. Most of the existing frequent subgraph mining algorithms deal with undirected and non-weighted graphs. Consideration of weights and direction to these networks requires highly complex analysis. However, in the real world for example, all the biological networks have directions associated. So directed and weighted graph mining is necessary.

The recent development of high-throughput technologies provides a range of opportunities to systematically characterize diverse types of biological networks. ‘Network Biology’ has been an emerging field in biology. Most of our world can be represented as networks including entities and relationship between the entities. For example, biological cells can be represented as biological networks, which include various molecules and relationships between molecules. With a large amount of data becoming available about biological networks in different species, the need of data mining for such networks is rapidly growing now days. There are several challenging problems in the analysis of biological networks, such as finding biologically meaningful patterns to help us to discover common motifs of cellular interaction, evolutionary relationships etc.

We model social, biological networks by weighted directed graphs, which can represent different entities from people, gene/protein, chemicals etc. as vertices and the relations between elements as directed edges. Now we can convert mining problems in any network into graph mining problem. Directed weighted graphs can often show more explicit and higher quantity of information than undirected unweighted graphs. Mining frequent pattern for directed weighted graphs can provide more useful knowledge or information. These networks are of large size, and discovering pertinent paths from these networks involves a computational process. In this study a novel sub-network mining algorithm was developed to find the most significant pathways from a weighted directed network.

1.5 The proposed model

In the proposed model, we describe a novel graph mining algorithm to discover significant and meaningful subnetworks and Maximal paths from a given network. This algorithm can mine subnetworks and Maximal paths from a weighted directed graph. Most of the existing algorithms are based on the topological structure of a network, such as node connectivity. Even though some algorithms take edge weight into consideration, the significance of the node is judged using the topology of the network alone. Moreover, none of algorithms deal with all the three parameters: node weight, edge weight, and direction of the edges. Hence, we developed a novel algorithm which can incorporate node weights, edge weights, direction, and obtain significant subnetworks and Maximal paths based on a user's interest. This algorithm allows the user to incorporate any numerical values of node weights and edge weights, and then mine the different subnetworks and Maximal paths from the network based on the node weight, edge weight, and its direction. The significance of the algorithm is that the user can create a network from the available data and then mine meaningful Maximal paths from it. The algorithm we have proposed reduces the complexity of finding canonical matrix from $O(n!)$ to $O(nlogn)$ in the best case and $O(n^2logn)$in the worst case.

CHAPTER TWO: RELATED WORK

There are several previously developed tools for querying paths and subnetworks from networks. Much work been done in the area of graph based mining and pathway discovery were on undirected and unweight graph, while none of them have the functionalities of the proposed algorithm.

Albert and Barabasi [3], Dorogovtsev and Mendas [38] have given extensive review on models of growing graphs. Newman [105] has given a shorter review by taking other view points and Hayes [62,63], who concentrate on the small-world models, and Strogatz [145], who includes an interesting discussion on the behavior of dynamical systems on networks. The book by Newman et al. [104] is a collection of previously published papers, and also contains some reviews given by editors. Albert-Laszlo Barabasi's focusing particularly on Barbasi's work on scale-free networks gives a personal account of recent developments in the study of networks. Within graph theory, the books by Harary [60] and by Bollobas [17] are widely cited, and among social networks theorists the books by Wasserman and Faust [154] and by Scott [129] are widely cited. The book by Ahuja et al. [2] is a useful source for information on different network algorithms. Work in the field of graph theory was inspired by a groundbreaking 1998 paper by Watts and Strogatz [156], gives a comparative study of networks from

different branches of science, with emphasis on properties that are common to many of them.

2.1 Background on networks

2.1.1 Social networks

Some of the ground breaking works on social networks include; Jacob Moreno's work in the 1920s and 30s on friendship patterns within small groups, the 'southern women study' of Davis et al. [35] 1936 which focused on the social circles of women in an unnamed city in the American south, the study by Elton Mayo and colleagues of social networks of factory workers in the late 1030s in Chicago [126], the mathematical models of Anatol Rapoport [120], who was one of the first theorists, perhaps the first to stress the importance of the degree distribution in networks of all kinds, and the studies of friendship networks of school children by Rapoport and the others [120].

Another important set of experiments are the famous "small-world" experiments of Milgram [97] that have analyzed a network of telephone calls made over the AT&T long distance network on a single day. Ebel et al. [44] have reconstructed the pattern of email communications between five thousands students at Kiel University from logs maintained by email servers. Email networks have also been studied by Newman et al. [107] and by Guimera et al. [59]; Smith [136] constructed similar networks for an instant messaging system; and for an internet community web site by Holme et al. [64]; Dodds et al. [37] have carried out an email version of Milgram's experiment in which participants

were asked to forward an email message to one of their friends in an effort to get the message ultimately to some chosen target individual.

2.1.2 Information networks

The classic example of an information network is the network of citations between academic papers. Another information network World Wide Web has been very heavily studied since its first appearance in the early 1990s, with the studies by Albert et al. [4,15], Kleinberg et al. [82], and Border et al. [19] being particularly influential. The network of relations between word classes in a thesaurus has been studied by Knuth [83] and more recently by various other authors [82,101,144]. A number of other semantic word networks have also been investigated [39,25,134,144].

2.1.3 Technological networks

Noted work on technological networks includes; statistical studies of power grids by, Watts and Strogatz [155,156] and Amaral et al. [6]. Studies of Internet structure have been carried out by, among others, Faloutsos et al. [48], Broido and Claffy [20] and Chen et al. [29]. Other distribution networks that have been studied include the network of airline routes, and networks of roads [74], railways [88,130] and pedestrian traffic [30]. River networks could be regarded as a naturally occurring form of distribution network [37,94,125].

2.1.4 Biological networks

Studies on biological networks include; statistical properties of metabolic networks by Jeong et al. [71], Fell and Wagner [51,152], and Stelling et al. [143]. Protein interaction networks have been studied by a number of authors [69,70,95,140,149]. The statistical structure of regulatory networks has been studied recently by various authors [50,158,132]. The work on random Boolean bets by Kauffman [75,76,77] is a classic in this field. Statistical studies of topologies of food webs have been carried out by Sole and Montoya [98,138], Camacho et al. [23] and Dunne et al. [41,42], among others. A particularly thorough study of webs of plants and herbivores has been conducted by Jordano et al. [72]. A best known work on neural networks is the re-construction of the 282-neuron neural network of the nematode C. Elegans by White et al. [158]. The network structure of the brain at larger scales than individual neurons functional areas and pathways has been investigated by Sporns et al. [141,143].

2.2 Graph based mining

Several topics in the research field are closely related to graph mining, but having a different focus. Relational data mining by Dzeroski and Lavrac [43] which uses the structure of linkage between multiple relations for finding patterns from database has attracted a lot of research interest recently. For instance, given a database of movies, actors, awards, and the labeled links between them (i.e. a graph), McGovern and Jensen [96] find the patterns (subgraphs) associated with predicting which movies will be nominated for academy awards every year. Relational learning typically focuses on finding small patterns at the local level while Graph based mining looks at the global

structure. The idea of mining frequent pattern was first introduced by Agrawal and Srikant [1], which follow the general principle of Apriori algorithm for association rule mining.

The graph based mining algorithm can be categorized into 5 groups. These include greedy search based algorithms, inductive logic programming (ILP) based algorithms, inductive database based algorithms, mathematical graph theory based algorithms and kernel function based algorithms.

SUBDUE [33] and GBI [162] are the two greedy search based algorithms which appeared around 1994. SUBDUE [33] which deal with conceptual graphs which belong to a class of connected graphs. The other one is called Graph Based Induction GBI [162] which was originally intended to find interesting concepts from inference patterns by extracting frequently appearing patterns in the inference trace falls. To our knowledge, the first system that tried complete search for the wider class of frequent substructure in graphs named WARMR was proposed in 1998. They combined ILP method with Apriori-like level wise search to a problem of carcinogenesis prediction of chemical compounds. To alleviate this difficulty, a new system called FARMAR has recently been proposed. FARMER also uses the level-wise search, but apply lesser strict equivalence relation under substitution to reduced atom sets. A work in the framework of inductive database, having practical computational efficiency is MolFea system based on the level-wise version space algorithm. This method performs the complete search of the paths embedded in a graph data set where the paths satisfy monotonic and anti-monotonic measures in the version space.

The mathematical graph theory based approach mines a complete set of subgraphs under mainly ‘support’ measure. The initial work is AGM [68] (Apriori-based Graph Mining) system. The basic principle of AGM is similar to the Apriori algorithm for basket analysis. Starting from frequent graphs where each graph is a single vertex, the frequent graphs having larger sizes are searched in bottom up manner by generating candidates having an extra vertex. Frequent Subgraph discovery system which also takes similar definition of canonical labeling of graphs based on the adjacency matrix. DFS [15] (Depth first Search) based canonical labeling approach called gSpan [160] (graph-based Substructure pattern mining) has been proposed. By applying this DFS coding and DFS search, gSpan can derive complete set of frequent subgraphs over a given minimum support in a very efficient manner in both computational time and memory consumption.

2.2.1 Graph based mining on biological networks

The growing interest in network biology has led to the need for advanced computational methods for network analysis and as a result, several tools have been developed. Metacore^{TM} [108] is a visualization tool which can be used for biomarker identification, network construction, path identification etc. Cytoscape [31] is another visualization-based software tool for constructing biological networks. Micro array data integration, GO-term enrichment analyses are some of the plugins offered by Cytoscape. VisANT [66] provides functional and topological analysis of nodes whereas Osprey [18] focuses on visualization. Another notable tool IsAViz [9], build on AT&T Graphviz [9] is specifically designed for visualization. BioPIXIE [21] is a gene-based query engine for pre-computed networks for Saccharmyces cervisiae. NetworkBLAST [73] allows a user

to compare two networks of different species using a similarity measure. GraphWeb [122] is another software which is designed to analyze individual or multiple merged networks, module discovery, and discover novel candidates. MATISSE [150] is useful for mapping high-throughput datasets onto network topologies and detecting gene modules using a number of algorithms. BiologicalNetworks [10] is a network retrieval, construction and visualization tool with an emphasis on microarray data. PathwayAssistant [115] is another tool which provides computational tools for metabolic modeling tasks.

Clustering is perhaps the most common approach for biological network analysis, and is frequently applied to uncover functional modules and protein complexes, and to infer protein function Bader and Hogue, [9]; Hartwell et al. [61]; Pereira-Leal et al. [117]; Rives and Galitski [123]; Spirin and Mirny [140]). As a result, numerous clustering algorithms for biological networks have been developed like Altaf-Ul-Amin et al. [5]; Bader and Hogue [9]; Blatt et al. [16]; Chen and Yuna. [28]; Colak et al. [32]; Enright et al. [46]; Georgii et al. [57]; King et al. [80]; Loewenstein et al. [91]; Navlakha et al. [103]; Palla et al. [112]; Samanta and Liang [127]; Sharan et al. [131].

Figure 1 shows some of the related work on subnetwork and pathway analysis. There exists several tools for querying biological networks including, Network alignment tools, Graemlin [53] by A Novak et al. PathBlast [78] by BP Kelly and Network blast [73]) which align protein-protein interaction networks by combining interaction topology and protein sequence similarity to identify conserved pathways. Network alignment has also been applied to metabolic networks [118]. Several tools exist for

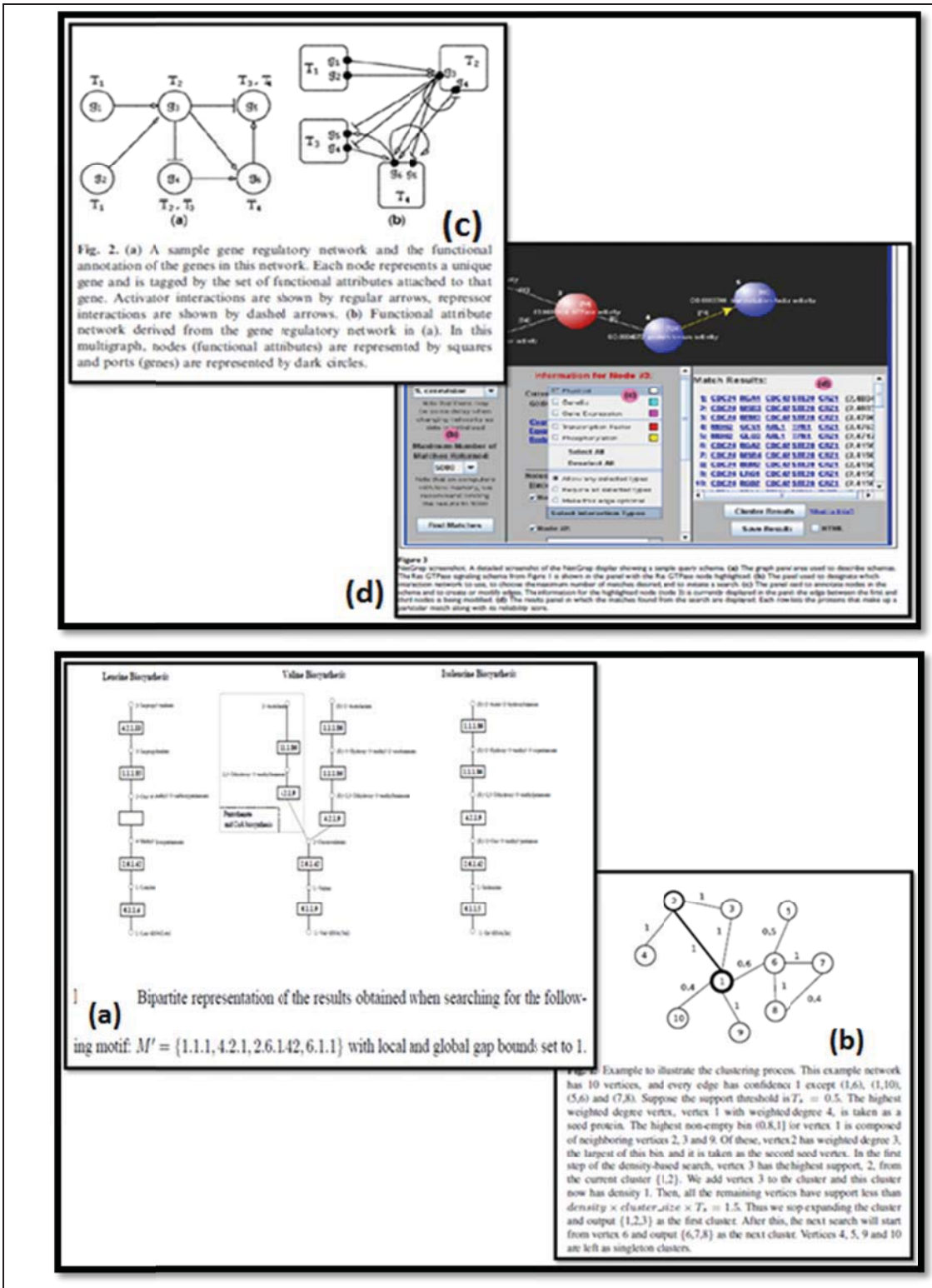


Figure 1: Some results from previous studies. Note: (a),(b),(c),(d) obtained from [21], [37], [49], [7] respectively.

uncovering network motifs or over-represented topological patterns in graphs like Fanmod [157] from S Wernicke and F Rasche, MAVisto.

The study of Saccharomyces Cervisiae transcription regulation network with a view to understanding relationships between functional categories was done by Lee et al. [89] Functional annotations of regulatory pathways [113] by M. Singh is another significant work in this area. NetGrep [11] by E Bank et al. is another system for searching protein interaction networks for matches to user-supplied 'network schemas'. In previous genome-scale studies [71], graphs have been used mainly for topological analyses regardless of the nature of their components. V. Lacroix et al. [87] studied motif search in graphs unlike other studies where topological features were considered along with other factors. QNet [40] by B Dost et al. is a tool that is used for querying pathways from a network.

Kelley et al. [78] devised an algorithm for querying linear pathways in PPI networks. Pinter et al. enabled fast queries of more general pathways that take the form of a tree. Their algorithm is limited to searching within a collection of trees rather than within a general network. Sohler and Zimmer [137] developed a general framework for subnetwork querying, which is based on translating the problem to that of finding a clique in an appropriately defined graph. Qpath [133] is used for identifying subnetworks of simple topology in a network. Another work in this area includes tYNA [161]. Cytoscape plugin that can submit a network to a remote server for detection of four common motifs and the MAVisto [128] by Schreiber et al. software finds over represented network motifs of a user defined size. NetMatch [52] by A. Ferro et al. is an

efficient graph matching algorithm with extensions to handle multiple labels per node, multiple edges between pairs of nodes, and approximate queries.

We can see that most of the works are on unweighted undirected graphs since weight and direction will make the whole process complicated. Even though there exist several works on weighted graph, it is only possible to incorporate weight on the edges. The subnetwork discovery is topological. Hence, a novel algorithm is developed that considers node weight along with edge weight and direction. Adding node weight can incorporate more knowledge to the network and helps in finding subnetworks and Maximal paths without considering topology. As in case of biological networks like Transcriptional regulatory networks, Signal transduction network, and Metabolite networks the edges are directed which add a lot more meaning to these networks. The weights associated with edges differentiate them in terms of strength, intensity or capacity which makes the network more expressive. The algorithm can derive significant subnetworks and Maximal paths with the help of node weight, edge weight and direction.

CHAPTER THREE: METHODOLOGY

This research focus on the automatic discovery of meaningful Maximal paths and subnetworks from weighted directed networks. In this section we will look mainly into the steps involved in network mining algorithm. The first step is data preprocessing where we try to incorporate all the knowledge relevant to each domain as node and edge weights. The two main steps involved in the network mining algorithm are canonical adjacency matrix generation and subnetwork or Maximal paths discovery. This section also discusses the different ways in which the discovered pathways are ranked or scored.

3.1 Definitions

3.1.1 Directed network or directed graph

A directed graph is a graph whose edges have direction and are called arcs. Arrows on the arcs are used to encode the directional information: an arc from vertex A to vertex B indicates that one may move from A to B but not from B to A.

A directed graph is an ordered pair $G = (V, E)$, where V is a non-empty set called the vertices of G, and E, called edge, is a finite set of ordered pairs of vertices such that $E = (v1, v2)$ where $v1, v2 \in V$. A direction given from A to B means, A must be

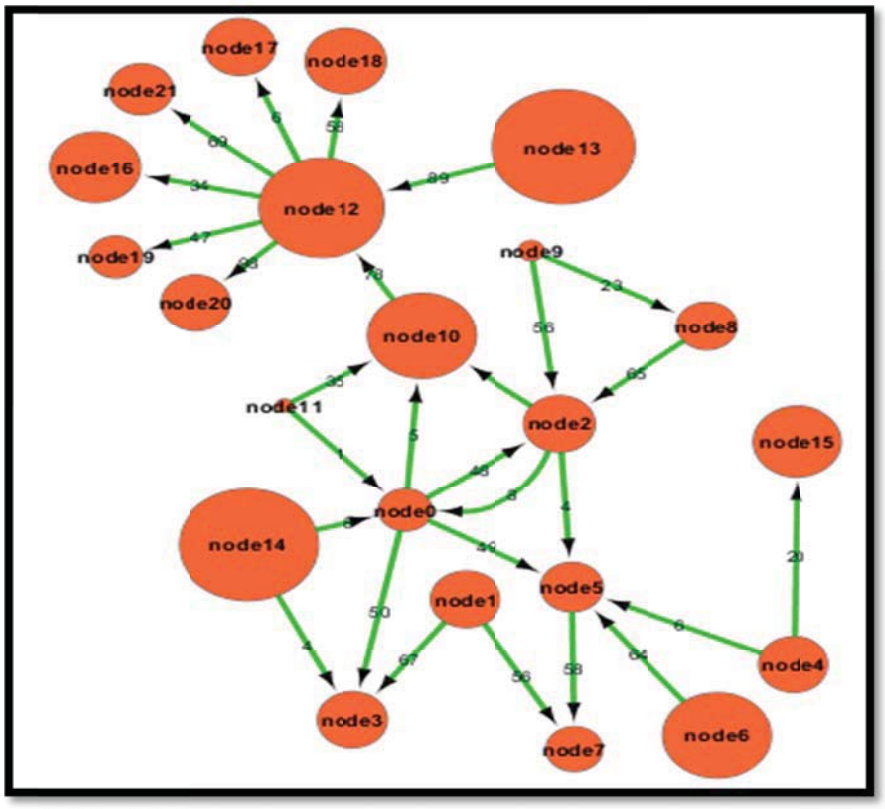


Figure 2: A weighted directed graph. Note: The size of the nodes represents the node weight and the value seen on the edges represents the edge weights. The arrow shows the direction of flow/interaction with the given graph/network.

completed before B starts. Using the approach of directed graph we can find out the features like reach ability, strong connectivity and more. An arc $e = (V1, V2)$ is considered to be directed from V1 to V2; V1 is called tail and V2 is called the head of the arc. If a path is made up of one or more successive arcs leads from V1 to V2, then V2 is said to be successor of V1, and V1 is said to be a predecessor of V2. The arc (V2, V1) is called arc (V1, V2) inverted. [15]

3.1.2 Weighted graphs

A weighted graph is a graph that has numerical label w(e) associated with each edge e, called the edge weight of the edge e. Each weight can be integers, rational numbers or real numbers which represent a concept such as distance, connections costs, or affinity.

Here are some good examples showing the importance of weighted graphs. Computer networks can be best represented using graphs. If we are interested in finding the fastest way to route a data packet between two computers then it is not appropriate for all the edges in the graph to be equal to each other. Likewise roads between cities can be best represented using graphs and if we want to find the fastest way to travel across the country. In this case too, it is not appropriate for all the edges to be equal to one another. Hence it is very important to consider graphs with edges which are not weighted equally. These have weights associated with the edges.

Apart from the edge weights, nodes can also incorporate valuable information or knowledge as edges do. For this reason, in this study we take node weight also into consideration. Similar to edge weights, node weight can be integers, rational numbers or real numbers. Weight of a node and an edge is represented by '$W_{N(i)}$' and '$W_{E(ij)}$' respectively where i and j represents two node names. Hence a weighted directed network is a graph with edge weights, node weights and direction.

Definition 1: Weighted graph: A weighted graph is a graph that has numerical label '$W_{N(i)}$' and '$W_{E(ij)}$' associated with each node and each edge, called the node weight of edge N(i) and edge weight of edge E(i,j) respectively.

3.1.3 Adjacency matrix

An adjacency matrix is a means of representing which vertices of a graph are adjacent to which other vertices. The adjacency matrix of a finite graph G on n vertices is the n × n matrix where the non-diagonal entry a_{ij} is the number of edges from vertex V_i to vertex V_j, and the diagonal entry a_{ii}, depending on the convention, is either once or twice the number of edges (loops) from vertex V_i to itself. Undirected graphs often use the former convention of counting loops twice, whereas directed graphs typically use the latter convention. There exists a unique adjacency matrix for each graph (up to permuting rows and columns), and it is not the adjacency matrix of any other graph.

In this study we represent adjacency matrix in a different format. Since we consider weighted graphs, the edge weights are entries to the matrix and the node weights/vertices represents the indices.

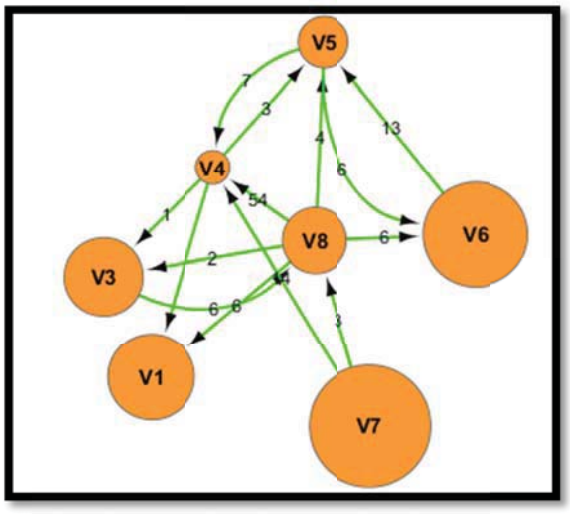
Definition 2: Adjacency matrix: Given a directed graph G=(V,E) with n vertices and m edges, the simplest graph representation schema use n x n matrix A as shown below :

$$A_{i,j} = \begin{cases} W & (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

'Wn(ij)' is the weight of an edge, i.e. the $(i ,j)^{th}$ element of a matrix is non-zero or Wn(ij), only if $V_i \rightarrow V_j$ is an edge in G. This kind of representation of matrix is called an adjacency matrix, as shown in Figure 2.

3.1.4 Weighted edges and nodes

A weighted graph is a special type of labeled graph in which the labels are numbers which are usually taken to be positive. Figure 1 is an example for weighted directed graph with edge weights, node weights and direction. The edges marked by numerical values represent the edge weight and the size of the node represent the node weights where higher the size greater the node weight.



a) A Weighted Directed Network

	V7	V6	V1	V3	V8	V5	V4
V7	0	0	0	0	**3**	0	**4**
V6	0	0	0	0	0	**13**	0
V1	0	0	0	0	0	0	0
V3	0	0	0	0	**6**	0	0
V8	0	**6**	**6**	**2**	0	**4**	**54**
V5	0	**6**	0	0	0	0	**7**
V4	0	0	**2**	**1**	0	**3**	0

b) Adjacency matrix of graph (a).

	V7	V6	V1	V3	V8	V5	V4
V7	0	0	0	0	**3**	0	**4**
V6	0	0	0	0	0	**13**	0
V3	0	0	0	0	**6**	0	0
V8	0	**6**	**6**	**2**	0	4	**54**
V5	0	**6**	0	0	0	0	**7**
V4	0	0	**2**	**1**	0	3	0

c) The Final adjacency matrix. Row V1 of 2(b) doesn't have any connection with other entries. Since the entire row of V1 is zero, it can be eliminated.

Figure 3: Adjacency matrix.

3.1.5 Graph isomorphism

In graph theory, an isomorphism of graphs G and H is a bijection between the vertex sets of G1 and G2.

f: V(G1)→ V(G2)

In a way that any two vertices u and v of G1 are adjacent in G1 if and only if f(u) and f(v) are adjacent in G2. Given a pair of graph G1 and G2, an isomorphism is a one to one mapping. In simple words, a graph is said to be isomorphic if every node and edge in G1 is also present in G2 with same adjacency.

3.1.6 Frequent subgraph mining or graph based mining

Graph mining and network analysis is critical to a variety of application domains, ranging from community detection in social networks, malicious program analysis in computer security, to searches for functional modules in biological paths and structural analysis in chemical compounds. There is an emerging need to systematically investigate the modeling, managing, and mining of large-scale graphs and networks in bioinformatics, social networks, and computer systems.

3.2 An overview

The objective of this work is to incorporate node weight, edge weight, and direction into a network and discover significant subnetworks and Maximal paths from them. The format of the methodology session is as follows: Section 3.3 discusses about the data preprocessing and network modeling; Section 3.4 discusses about transformation of adjacency matrix to canonical adjacency matrix; Section 3.5 discusses about

algorithms for canonical adjacency matrix; Section 3.6 discusses about Maximal paths or subnetwork generation; Section 3.7 discusses about Maximal paths ranking followed by section 3.8 Performance analysis.

3.3 Data preprocessing and network modeling

The first phase of methodology is data preprocessing. The edge weight $W_{E(ij)}$, node weight $W_{N(i)}$, and direction play a crucial role in subnetwork discovery. Hence the decision-making for selecting weights is an important though tedious task. Once all the weights have been determined, one can embed them in the network, thus modeling the network as a weighted directed network. Weight selection can transform from domain to domain upon user's interest.

Here are few examples for picking node weights and edge weights to model a network. Social network can be considered as a network where people interact with each other. For instance, Facebook is a case where the network can be viewed as a connection or interaction between friends and friends of friends. Upon user's interest it is possible to assign a numerical value to a node (node weight $W_{N(i)}$) and an edge (edge weight $W_{E(i,j)}$) such that it can incorporate sufficient knowledge. Consider a case where the user's interest is to find the most active user's on Facebook and the pattern in which they interact with each other. In this case it is significant to assign node weight as degree, since degree describes the connectivity with other nodes in the network. Also, the amount of information exchanged between friends (scraps/chats/messages etc.) taken as a numerical value can be assigned to edges as edge weight. This is how a weighted directed network is modeled.

Another example is a rumor mill where the user's interest is to find the most trusted source and the flow of rumor. In this case, the node weight can be a numerical value indicating a trusted person such that higher the value more trusted is the individual. Similarly edge weights can be the amount of rumor they exchange.

Similarly in a study of a biological network it is an efficient way to consider node weight as degree, closeness, expression score etc., and edge weight as gene ontology distance, PPI interaction score and more.

This weight decision is very crucial as the subnetwork or Maximal path generation relay entirely on node weights, edge weights and direction. Once the weight has been decided then the data can be modeled as a weighted directed network. Next, the data can be filtered using a user defined threshold 'σ', This is meaningful in the context of a domain that helps in reducing the network size by ignoring the less frequent nodes and less frequent edges.

3.3.1 Node parameters

This section shows a detail description of some of the edge and node parameters which can be adopted for modeling a network. Within graph theory and network analysis, there exist various measures of centrality of a node/vertex within a graph that determines the importance of the node within the graph. For instance; 'how important is a person within social network', 'how important a room in a building' or 'how well a road is used with in an urban area' [147]. The below listed are some of the measures that can be adopted as node weights.

Degree:

Degree is defined as the number of links incident to a particular node. Degree is often interpreted in terms of the immediate risk of node for catching whatever is flowing through the network such as a virus, or some information. Indegree is a count of the number of links directed to the node, and outdegree is the number of links that the node directs to others. For positive relations such as friendship or advice, we normally interpret indegree as a form of popularity, and outdegree as gregariousness. [148]

The degree sum formula states that, given a graph G = (V,E). [36]

$$\sum_{v \in V} \deg(v) = 2|E| .$$

Betweenness:

Betweenness is a centrality measure of a node within a graph. Vertices that occur on many shortest paths between other vertices have higher betweenness than those that do not.

For a graph G = (V,E) with n vertices, the betweenness $C_B(v)$ for vertex v is computed as follows:

1. For each pair of vertices (s,t), compute all shortest path between them.

2. For each pair of vertices (s,t), determine the fraction of shortest paths that pass through the vertex in question (here, vertex v).

3. Sum this fraction over all pairs of vertices (s,t).

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

Where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through a vertex v. [54]

Closeness:

Vertices that are shallow to other vertices i.e. those that tend to have short geodesic distances to other vertices with in the graph have higher closeness. Closeness is preferred in network analysis to mean shortest-path length, as it gives higher values to more central vertices, and so is usually positively associated with other measures such as degree.

$$\frac{\sum_{t \in V \setminus v} d_G(v,t)}{n-1}$$

where $n \geq 2$ is the size of the network's 'connectivity component' V reachable from v. [148]

Eigen Vectors:

It is a measure of the importance of a node in a network. It assigns relative scores to all nodes in the network based on the principle that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. Google's PageRank is a variant of the Eigenvector centrality measure.

Let x_i denote the score of the i^{th} node. Let $A_{i,j}$ be the adjacency matrix of the network. Hence $A_{i,j} = 1$ if the i^{th} node is adjacent to the j^{th} node, and $A_{i,j} = 0$ otherwise. More generally, the entries in A can be real numbers representing connection strengths, as in a stochastic matrix.

For the i^{th} node, let the centrality score be proportional to the sum of the scores of all nodes which are connected to it. Hence

$$x_i = \frac{1}{\lambda} \sum_{j \in M(i)} x_j = \frac{1}{\lambda} \sum_{j=1}^{N} A_{i,j} x_j$$

where M(i) is the set of nodes that are connected to the i^{th} node, N is the total number of nodes and λ is a constant. In vector notation this can be rewritten as

$\mathbf{x} = \frac{1}{\lambda}\mathbf{A}\mathbf{x}$, or as the eigenvector equation $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$. [148]

Clustering coefficients:

A measure of the likelihood that two associates of a node are associates themselves. A higher clustering coefficient indicates a greater 'cliquishness'.

The clustering coefficient for the whole network is given by Watts and Strogatz as the average of the local clustering coefficients of all the vertices n. [148]

$$\bar{C} = \frac{1}{n} \sum_{i=1}^{n} C_i.$$

Reach:

The degree any member of a network can reach other members of the network. [148]

Structural Cohesion:

The minimum number of members who, if removed from a group, would disconnect the group. [148]

Structural Equivalence:

Refers to the extent to which nodes have a common set of linkages to other nodes in the system. The nodes don't need to have any ties to each other to be structurally equivalent. [148]

Second order centrality:

It assigns relative scores to all nodes in the network based on the observation that important nodes see a random walk (running on the network) “more regularly” than other nodes. [148]

3.3.2 Edge parameters

Measure of dissimilarity:

Edge weight can be a value which tells the measure of dissimilarity between two nodes N1 and N2. Larger the values more dissimilar are the nodes.

Statistical correlation:

It is the similarity measure between two nodes in a network. [24]

3.3.3 Biological parameters

All the parameters discussed in the section 3.3.1 and 3.3.2 is applicable for biological networks to consider as node weights and edge weights respectively. Apart from these, there exist other node parameters like gene expression value, PPI score, gene ontology distance etc. The edge weight Wij could be d(i,j), the shortest distance between i and j in G’. Other measures are k(i,j) and k’(i,j), which denote, respectively, the minimum number of vertices and edges that need to be removed from G’ to disconnect i and j. As these are the measures of similarity, we could obtain required edge weight as $Wij = |V'| - k(i,j)$ or $|E'| - k'(i,j)$. These structural properties of weights can be computed in polynomial time and are applicable to protein protein interaction networks. The statistical correlations between genes are applicable to gene co expression networks. Similarly gene

ontology distance and protein protein interaction score are the other two edge weight parameters.

3.4 Transformation to canonical adjacency matrix

Once the network is modeled then it is represented by an adjacency matrix. The entries in the adjacency matrix represent the edges $W_{E(ij)}$, and the indices represent the nodes $W_{N(i)}$. In order to avoid the isomorphism problem of graphs, we first transform the graph into a unique, canonical representation. We use a variation on the method of vertex invariants [8] in order to find the canonical adjacency matrix. This unique way of finding the canonical adjacency matrix makes the computational process much faster.

3.4.1 Canonical adjacency matrix

Graph isomorphism is one of the critical problems in graph mining. It is necessary to get rid of the problem of isomorphism.

In order to check whether two given graphs G1 and G2 are isomorphic we first transform each graph into an adjacency matrix Ad_1 and Ad_2 followed by a canonical adjacency matrix transformation and a comparison. Given a pair of graph G1 and G2, an isomorphism is a one to one mapping. In terms of adjacency matrix, two graphs G1 and G2 are isomorphic if a permutation of rows and corresponding columns in adjacency matrix is same as each other.

While developing the canonical adjacency matrix, the edge weights and node weights are considered and hence the canonical adjacency matrix CAM will be a unique matrix in which the nodes and edges are arranged according to their priority as we set.

Based on a threshold, in context of a domain, the entries to the canonical adjacency matrix CAM are limited. Weights above a particular threshold ‘σ’ will only be considered for canonical adjacency matrix.

To get the total order of graphs we use canonical labeling. A canonical label is a unique code representing a graph. Depending on the order of its edges E or vertices V, a graph G can be represented in many different ways. Nevertheless, canonical labels should always be the same despite how the graphs are represented, as long as those graphs have the same labeling of edges and vertices. Thus by using this unique canonical representation it is very efficient and easy to compare graphs because if two graphs G_1 and G_2 are isomorphic with each other, their canonical labels must be identical.

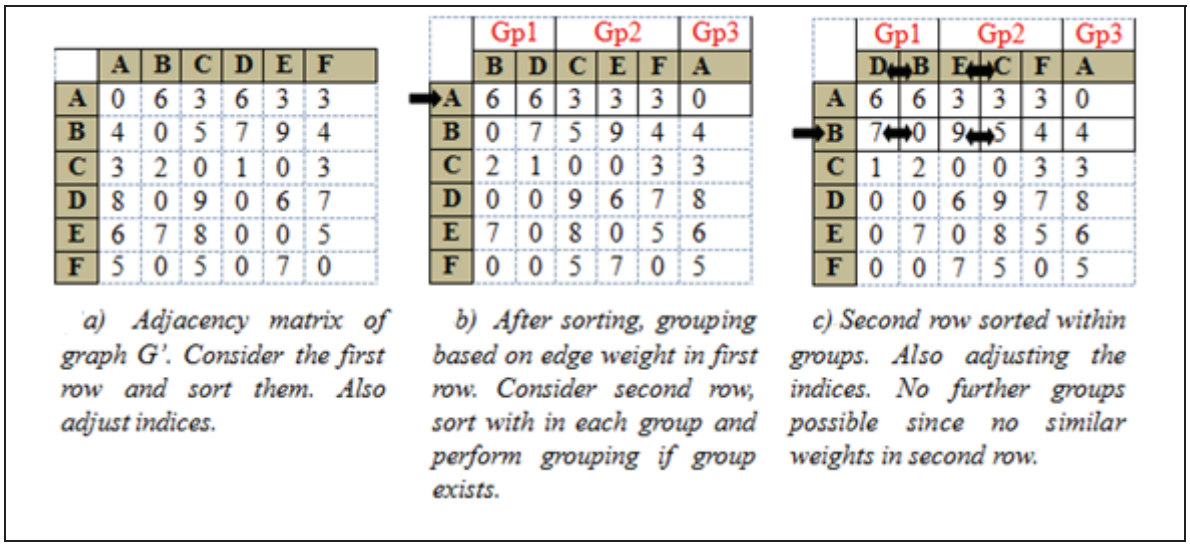
Apart from the traditional way of finding the canonical matrix by considering all permutations, which is time consuming and of high complexity, we use a variation to the method of vertex invariants where both vertex and edge are considered for partitioning and grouping, making it an efficient method, where as in traditional method only vertex will be considered.

3.4.2 The algorithm for canonical adjacency matrix

In this section, we formulate our algorithm and proof with an example. Algorithm uses adjacency matrix representation to store the graphs.

Step 1: (Line 1-3): All the vertices V of a given Graph G are sorted based on $W_{N(i)}$. Vertices which fall below the threshold ‘σ’ are strictly eliminated. This helps in reducing the input size also by eliminating the infrequent vertices. The adjacency matrix Ad_G is created based on the priority of vertices V. As shown in Figure 3a, the indices of the

matrix are arranged in accordance with the weight of the node such that 'A' has higher weight than 'B'. i.e. "$W_{N(A)}>W_{N(B)}>W_{N(C)}>W_{N(D)}>W_{N(E)}>W_{N(F)}$". Hence, it is not just a random adjacency matrix, but a matrix arranged on the basis of the



	A	B	C	D	E	F
A	0	6	3	6	3	3
B	4	0	5	7	9	4
C	3	2	0	1	0	3
D	8	0	9	0	6	7
E	6	7	8	0	0	5
F	5	0	5	0	7	0

a) Adjacency matrix of graph G'. Consider the first row and sort them. Also adjust indices.

	Gp1		Gp2			Gp3
	B	D	C	E	F	A
A	6	6	3	3	3	0
B	0	7	5	9	4	4
C	2	1	0	0	3	3
D	0	0	9	6	7	8
E	7	0	8	0	5	6
F	0	0	5	7	0	5

b) After sorting, grouping based on edge weight in first row. Consider second row, sort with in each group and perform grouping if group exists.

	Gp1		Gp2			Gp3
	D	B	E	C	F	A
A	6	6	3	3	3	0
B	7	0	9	5	4	4
C	1	2	0	0	3	3
D	0	0	6	9	7	8
E	0	7	0	8	5	6
F	0	0	7	5	0	5

c) Second row sorted within groups. Also adjusting the indices. No further groups possible since no similar weights in second row.

Figure 4: Canonical adjacency matrix generation. Note: By sorting, adjusting indices and then grouping. Gp1, Gp2, Gp3 are the three groups. After grouping, further sorting and grouping is done within each groups in the following row.

ALGORITHM 1: Canonicaladjmatrix(graph G')

1. Sort the labels W_{Ni} of vertices V in the graph by frequency;
2. Remove infrequent vertices W depending on the threshold 'σ';
3. Relabel the remaining vertices V in the descending order;
4. Create adjacency matrix Ad_x with V' as indices and weight W_{Ei} of directed edges as entries;
5. Call sort_adjust(int start, int end);
6. Return the canonical adjacency matrix CAM(G');

node weight $W_{N(i)}$. The nodes in the graph with higher weight are in the upper part of the matrix, this makes it useful while retrieving for subnetworks. Eliminate if any row entry or column entry is entirely zero. The row indices keep track of the nodes and the column

indices represented by vertices is used to keep track of the edges. Hence this adjacency matrix may not be a nxn matrix after eliminating an entire row or column with zero's if it exists.

Step 2: (Line 4): An adjacency matrix $Ad_{(G)}$ is created as shown in Figure 1, where x,y,z are frequent nodes. The edge weights $W_{E(ij)}$ of directed edges are taken as the entries into the matrix.

Step 3: (Line 5): After the adjacency matrix $Ad_{(G)}$ is created, a set of operations are done inorder to form a canonical adjacency matrix $CAM_{(G)}$. Function sort_adjust() is called to sort the edge weights $W_{E(ij)}$ and adjust the column indices accordingly such that the highest frequent edge moves to the left part of the matrix. We rearrange the edge weights $W_{E(ij)}$ while keeping track of node weight $W_{N(i)}$ (highly weighted nodes). This is done by changing only the column indices $W_{N(j)}$ while $W_{N(i)}$ remains the same as shown in Figure 4b and 4c.

Subprocedure 1: sort_adjust(int start, int end)

```
1.  for i=start to end
2.     for j=i to end
3.        if entry_{i,j} < entry_{i,j+1}
4.           update_index(i,j)
5.        end if
6.     end for j
7.  end for i
```

Figure 4 indicates operations on the adjacency matrix. Figure 4b. is the adjacency matrix $Ad_{(G")}$ developed from a given graph G". Now we apply a slight variation of method of vertex invariants [8]. We may call it edge invariants. As shown in Figure 4b,

the first row of adjacency matrix ($Ad_{(G")}[0][j]$) is sorted in descending order and the column indices $V_{N(j)}$ are adjusted. Sub procedure 1 and sub procedure 2 explains this.

Subprocedure 2: update_index(int old, int new)

1. Swap the values of indexarray
   ie. Changing the indices

As shown in Figure 4b. Considering the sorted first row where "$W_{E(AB)} > W_{E(AD)} > W_{E(AC)} > W_{E(AE)} > W_{E(AF)} > W_{E(AA)}$", the indices $W_{N(j)}$ are adjusted accordingly, and are separated as groups (in colors) if entries share same edge weights $W_{E(ij).}$

After grouping based on first row (first row indicates the highly weighted node in the network), the function sort_adjust() is called for the second row for individual group. i.e. individual groups $Gp_{(i)}$ in the second row ($Ad_{(G")}[1][j]$) is sorted in descending order and indices are adjusted and grouped again if further group possible. Then next group $Gp_{(i+1)}$ is taken and the process continues. There could be groups within a group if they have same edge weights in this case the same process is repeated.

Subprocedure 3: group(int start, int end)

1. for i= start to end
2. if a[indexarray[i]] != a[indexarray[i+1]]
3. Dynamicgroupdetail.add(i) // keep
   // track of the group by keeping track of //indices
4. end if
5. end for i
6. if # elements in indexarray= # elements in dynamicgroupdetailarray.
7. Stop iteration // Cam fixed

Step 1: (Lines 1 to 6): Steps one to five show how to keep track of the indices $W_{E(ij)}$ after grouping as shown in Figure 4c.

Step 2: (Lines 6 and 7): The process of grouping, sorting and adjusting the indices row by row continues until no further group exists.

After second row, third row is sorted, adjusted and grouped. This process continues when until no further group is possible within a group. Thus, rather than considering all the rows and columns and including permutations to find canonical adjacency matrix, it is enough to consider only few rows and just compare the values. This reduces the complexity compared to the traditional way.

To conclude, instead of comparing all the entries we need to consider only the initial rows which are already arranged based on the node weights. Hence the final matrix after all grouping and adjusting will be the canonical adjacency matrix CAM(G").

Suppose we have a graph with n vertices. By edge invariants, we can create partitions based on the edge weights, In the traditional way of finding canonical labeling the complexity is n! that is $O(n^2)$. By using this method the complexity of finding the canonical labeling is $O(n\log n)$ in the best case and $O(n^2\log n)$ in the worst case. The best case is where there are no groups possible in the first row itself. The worst case is having just one group, i.e. having all entries same in a row same as having only one group in the entire rows.

3.4.3 Maximal path or subnetwork generation

Once the canonical adjacency matrix is fixed, then there are different ways in which we can get the data out of these matrices, based on the nodes to be appeared in the

sub network. This varies according to the domains whether it is biological, social etc. This section formulates the algorithm for sub network retrieval.

Definition 3: Maximal path:

Maximal path is a pattern which shows the sequence of link connecting nodes. For example, A Maximal path $A \rightarrow B \rightarrow C \rightarrow D$ is a pattern or a flow by which 'A' communicates/interacts with 'D' such that 'A' is the source and 'D' is the destination.

P1 is called a Maximal path in network N, if and only if, all the nodes and edges in P1 ϵ N and degree d of node; d $<= 2$.

As shown in Figure 4, there are eight different ways in which the sub networks can be generated as upon the user's interests and with context of domain.

ALGORITHM 2: Automatic(canonical adj matrix)

```
1.    If case={ 1,2,3,4,6}
2.        Start with the first row or first column;
3.    If case= {1,3,7}
4.        Call Extension() function;
5.    If case={ 2,3,4,5}
6.        Enter the limits for rows and column; // user defined.
7.    Else If case=6
8.        Select the highest element from each row;
```

The sub network extraction starts with the first column or first row of the canonical adjacency matrix as it has the highest weighted in most of cases. Then function extension() is called where the network is expanded either edge wise or node wise. Thus, both the node and the edge can be expanded. Extension is the process of expanding the subnetwork either edge wise or node wise.

Figure 5 is a diagrammatic representation of subnetwork or Maximal path extraction. Each of the cases in Figure 5 are described below.

Subprocedure 1: Extension(canonical adj matix A)

1. If $(i + 1) ! = n$ and $(j + 1) ! = m$;
2. For i+1 to n and j+1 to m;
3. Increment i as i++ and j as j++;
4. End if; end for i; end for j;

Case 1:

As shown in Figure 5a. This is to find the most frequent node and its interconnection in the network thus forming a new subnetwork. We can extend this by adding the next frequent node in the network and its interconnections forming one more subnetwork. Then in Figure 5a, the next node that adds to the network is when we extend from node 'B' to node 'C' and so on.

$$\text{Output} = (\text{Ad}(\text{G''})[\rho][j])\ \rho = 0;\ j = 0 \text{ to } m.$$

$$(\text{Ad}(\text{G''})[\rho][j])\ p = \{0..n\};\ j = 0 \text{ to } m \text{ if extension.}$$

Case 2:

As shown in Figure 5b, if the user want a definite/particular number of frequent nodes and edges to appear in the subnetwork. In Figure 5c the first four frequent nodes and three frequent edges appear in the subnetwork. This can also be further extended by adding a new node or a new edge at a time.

$$\text{Output} = (\text{Ad}(\text{G''})[i][j])\ i = 0 \text{ to } x;\ j = 0 \text{ to } y;\ x = \{0,1 \ldots . n), y = \{0,1 \ldots m).$$

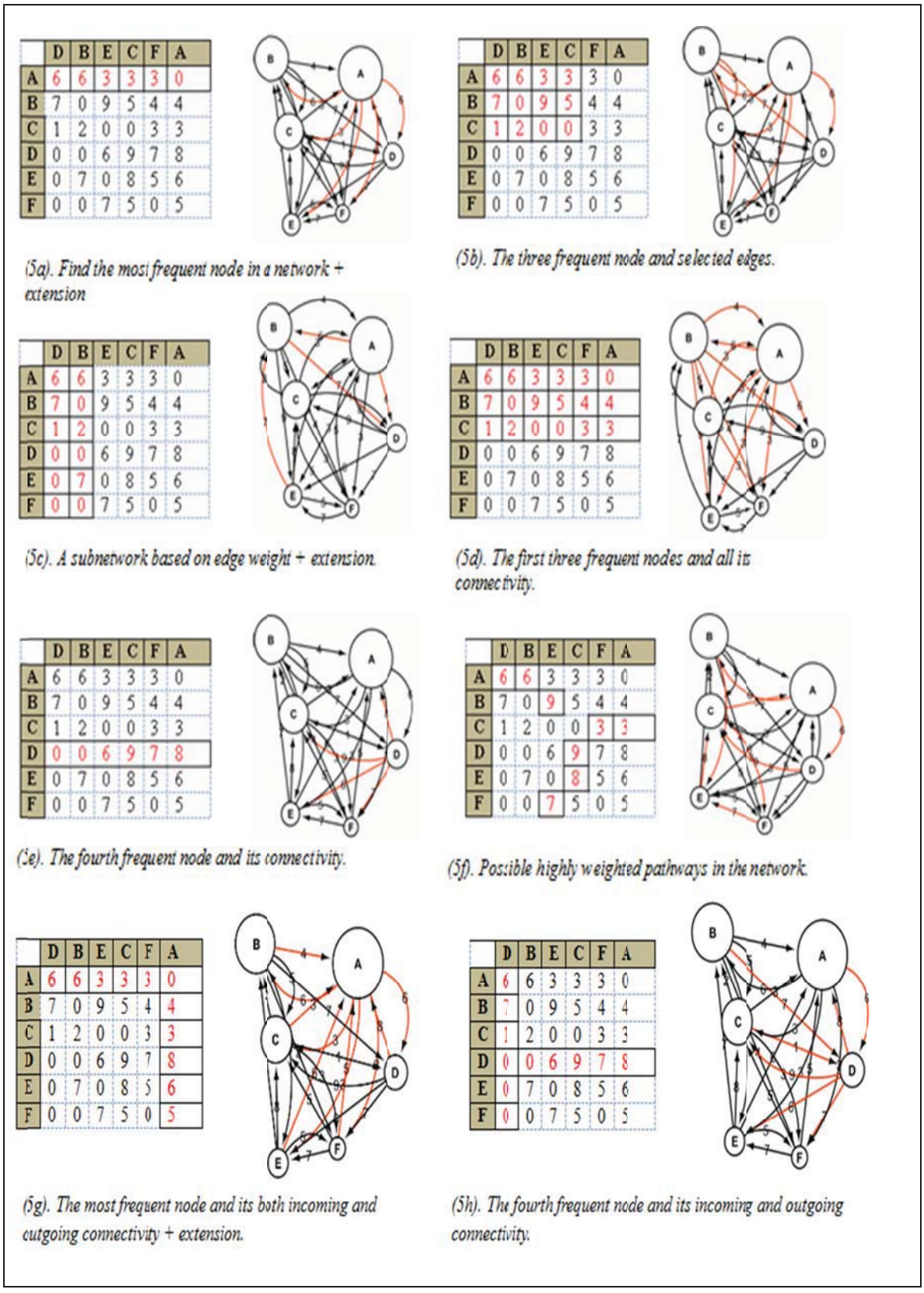


Figure 5: The different ways of sub network generation.

Case 3:

As shown in Figure 5c, retrieval by column. Subnetworks are created based on the edge weight.

$$\text{Output} = (\text{Ad}(\text{G''})[\text{i}][0])\ \text{i} = 0 \text{ to n.}$$

$$(\text{Ad}(\text{G''})[\text{i}][\text{j}])\ \text{i} = 0 \text{ to n};\ \text{j} = 0 \text{ to m if extension.}$$

Case 4:

Figure 5d, if we want to find for a set of node without extending one by one, It is possible to enter the number such that all frequent nodes will appear in the subnetwork. As shown in Figure 5d, if the user enters a number as three, then all frequent nodes and its interconnections upto level three will appear in the subnetwork.

$$\text{Output} = (\text{Ad}(\text{G''})[\text{i}][\text{j}])\ \text{i} = 0 \text{ to x};\ \text{j} = 0 \text{ to m};\ \text{x} = \{0,1 \ldots . \text{n}).$$

Case 5:

As shown in Figure 5e, Finding the nth frequent element and then extending after that. Hence at one step expansion the 6^{th} frequent element will appear in the subnetwork and so on.

$$\text{Output} = (\text{Ad}(\text{G''})[\text{i}][\text{j}])\ \text{i} = \text{x};\ \text{j} = 0 \text{ to m};\ \text{x} = \{0,1 \ldots . \text{n}).$$

Case 6:

As shown in Figure 5f. Find the most frequent Maximal path a node X to node Y. In this case canonical adjacency matrix is useful in retrieving the nodes in decreasing order of node weights one after the other forming the start of new path. This reduces the search time. The highest edge weight among the frequent nodes will be considering in creating the path. From Figure 5f, the Maximal paths discovered are $\text{A} \rightarrow \text{B} \rightarrow \text{E} \rightarrow \text{C} \rightarrow \text{A}$;

$$A \rightarrow B \rightarrow E \rightarrow C \rightarrow F \rightarrow E;\ A \rightarrow D \rightarrow C \rightarrow A;\ A \rightarrow D \rightarrow C \rightarrow F \rightarrow E \rightarrow C;$$

$$\text{Output} = \max(\text{Ad}(G'')[i][j])\ i = 0 \text{ to } n;\ j = 0 \text{ to } m;$$

Case 7 and Case 8:

In the above cases, when we start from frequent or highly weighted node, we only considered the outgoing edges from a node, but based on the study it was found that it is interesting to know the incoming edges too. Thus, in the case 7 and case 8, as shown in Figure 5g and 5h we can find the subnetwork such that from a node we can find out the incoming and outgoing edges.

$$\text{Output(Figure 5g)} = (\text{Ad}(G'')[0][j])\ j = 0 \text{ to } m.\ \text{Ad}(G'')[Vj][j] \text{ for } Vj = Vi \text{ and } j = 0 \text{to } m.$$

$$(\text{Ad}(G'')[i][j])\ i = 0 \text{ to } n;\ j = 0 \text{ to } m;\ \text{Ad}(G'')[Vj][j] \text{ for } Vj = Vi \text{ and } j = 0 \text{to } m \text{ if extension.}$$

3.5 Maximal path ranking

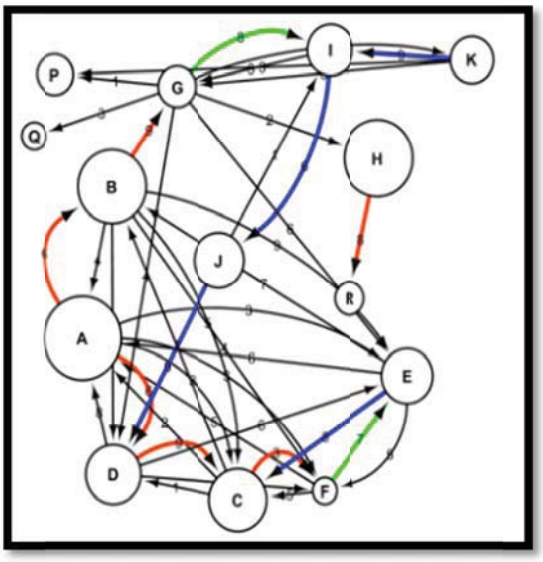
A Maximal path is the direction showing the flow of information. As a result of the case 6 of the algorithm, it is possible to find out the significant Maximal paths from a given network. But it is relevant to rank these paths. A subnetwork is said to be a Maximal paths if it is starting from a node and the edges connecting to the next node are in the same direction. i.e. it should show a Maximal path from one node to the another. In this case a Maximal paths can be user defined based on each domain, means an output is said to be a Maximal paths if it shows at least β edges in the same direction starting from a node connecting other nodes. Where $\beta = [3,4 \ldots . n)$, n is the number of rows in the

canonical adjacency matrix. An edge is selected such that the starting from a node, the highly weighted outgoing edge from that node is considered.

For example, In Figure 6b, $A \rightarrow D \rightarrow C \rightarrow F \rightarrow E \rightarrow C$, $A \rightarrow B \rightarrow G \rightarrow I \rightarrow J \rightarrow D$ $K \rightarrow I \rightarrow J \rightarrow D \rightarrow C \rightarrow F$ are the highly weighted Maximal paths where β=4. There exist several other Maximal paths other than these in the network. But the algorithm will find only the important Maximal paths based on the edge weight, node weight and direction and then rank them. How are these Maximal paths ranked?

	D	B	E	F	C	G	A	R	I	J	K	Q	H	P
A	6	6	3	3	3									
B	7	0	5	4	4	9	4							
H								8						
C	1	2		2			3							
D			6	7	9		8							
E		7		5	8		6		1					
J	9													
I						3				6				
K									9					
G	4		6						8		3	3	2	1
F			7		5		5							

a) A Canonical adjacency matrix divided into three parts as shown in three different colors. The colored entry is the edge selected for the paths.



b) A network showing different Maximal paths discovered using the algorithm. The color indicates the significance of the edges.

Figure 6: Maximal paths ranking.

There are three different ways in which the Maximal paths are ranked.

(i). Considering all the edges and nodes from a discovered path. This is called as Path score and is defined as follows.

Path Score:

Each subnetwork or Maximal path that is discovered is scored using the node weight and edge weight using the following formula,

$$\text{Path score} = (\sum(W(N)) + \sum(W(E))) / (N + E)$$

where N and E are the total number of nodes and edges in the path, $W(N)$ is the weight of each node, and $W(E)$ is the weight of each edge in the path. Thus, every Maximal path is ranked based on the score. A Maximal path with the highest score has the highest rank in the overall network.

(ii). Comparison of each node and edge from the discovery Maximal paths to the literature and then score the path. This is called as Accuracy rate and is defined as follows.

Accuracy Rate:

The accuracy rate is the correctness of the Maximal paths in comparison with similar Maximal paths from literature. It is measured on scale from 0 to 1. If both of the Maximal paths are exactly similar then accuracy rate is 100%; i.e. 1. This is measured as,

$$\text{Accuracy rate} = ((N + E) - (MN + ME))/ (N + E)$$

where N and E refers to the total number of nodes and edges in a derived path. And M_N and M_E refer to the total number of missing nodes and missing edges respectively in comparison with a standard Maximal paths from literature.

(iii). Edges and nodes below a particular user defined threshold are eliminated from the Maximal paths and then score the path.

The canonical adjacency matrix can be divide into each sections of η (selected based on each domain) where $\eta >= \beta \& <= n$; n is defined as the total number of

rows in the cam. In Figure 6a, we can see η=5, hence the cam is divided into equal parts of 5 rows each. Hence the first five rows have higher significance than the next five because of the higher value of edge weight. Similarly any Maximal paths identified using the first set of rows will have higher significance than the Maximal paths found in the next set of rows.

As shown in Figure 6b, the edge color indicates each section in the canonical adjacency matrix, such that red for the first set, blue for next and then green. Hence red edges will have higher significance than the blue and then comes the green. Similarly any Maximal paths with more number of red edges can be ranked higher. Maximal paths $A \rightarrow D \rightarrow C \rightarrow F \rightarrow E \rightarrow C$ can be ranked higher than Maximal paths with two red colored edges like $A \rightarrow B \rightarrow G \rightarrow I \rightarrow J \rightarrow D$.

Score:

The score for the Maximal path can be defined as,

$$\text{Score} = \left(\sum\left(\text{W(N)}\right)\right) \text{ where } \text{W(N)} \geq \beta$$

where W(N) is the weight of each node.

3.6 Performance analysis

In the traditional way of finding canonical labeling, the complexity is O(n!). The algorithm we have proposed here reduces the complexity to O(nlogn) in the best case and $\text{O}(\text{n}^2\text{logn})$ i in the worst case; the best case being where there are no groups possible in the first row itself, and the worst case is when there exists just one group, i.e. having all entries the same in a row is the same as having only one group in the entire row.

CHAPTER FOUR: EXPERIMENT RESULTS

In this section, we report a systematic performance study on synthetic datasets and real time datasets. The experiments were conducted on a computer with a Pentium M 1.4 G CPU and 512 M main memory, running Microsoft Windows XP operating system. We implemented the algorithm in C and VB.net. We used several synthetic data sets to test the algorithms.

4.1 Synthetic datasets

We used synthetic datasets to test the performance of the algorithm. Several facts were taken into consideration while creating the network. Edge weight, Node weight and Direction were randomly assigned to the network. A network with size of 50 nodes and 110 edges were considered to test the performance. We randomly assigned different set of weight to the same network and found out different pathways and subnetworks.

Table 1 is an analysis on different kinds of synthetic networks. It shows the possible kind of questions/queries related to the 8 cases mentioned in the methodology section. In addition, it also shows that the algorithm can be applied to distinct kinds of networks like social networks, biological networks etc. in order to retrieve information about connectivity, relevance of nodes, hub property etc.

4.1.1 A social network

This section is a detail demonstration of a complete experiment on social network along with the visualization of the subnetworks and paths. Consider a social network (Facebook) where people communicate with each other and exchange information. As mentioned above, the 50 nodes can be 50 people in Facebook and 110 edges are the interaction between them. Let the node weight be a value signifying the fame of a person. Higher the node weight more famous is the individual. The edge weight is the measure of common interests between people. Thus forming a meaningful network with some knowledge incorporated into it by means of edge weight and node weight and direction. The algorithm was run on this network resulting in several subnetworks. Here are some visualization of the results.

In all the displayed cases where does edge weight play a role? By using the edge weight we can eliminate some of the connections from the subnetwork. If the query is "Find the two most famous people from the group and the people they communicate where the common interest is greater than 2". In this case, from Figure 7 the edge Flo → Ura and Fay → Ira will be eliminated. This is because the edge weight is not greater than 2.

Table 1: An analysis of different networks.

Case 1:	**Finding the most frequent node and it connectivity**
Rumor mill	Who is the most trusted source and to whom all he spread the rumor?
Disease network	Which is the most deadly disease and what all other disease it can cause?
Cancer network	Which is the most frequent gene based on weight assigned?
Facebook	Who is the most active member and to whom all he converse?

Case 2:	**Finding the most n frequent nodes and its connectivity**
Rumor mill	Find 3 trusted source and its connectivity?
Disease network	Find 4 deadly disease and what all other disease it can cause?
Cancer network	Find the 3 active genes?
Facebook	Find the most 3 active members?

Case 3:	**Sub network based on edge weight**
Rumor mill	Find the Maximal path through which maximum flow of information?
Disease network	Find the disease which share maximum common symptoms?
Cancer network	Find a subnetwork?
Facebook	Find people who share most common interests?

Case 4:	**Subnetworks based on a set of node and set of edges**
Rumor mill	Find 3 trusted source and 4^{th} level of information passed?
Disease network	Find 2 mostly contagious disease and 3 levels of common symptoms?
Cancer network	Find 4 frequent genes and 3levels of frequent interconnections?
Facebook	Find 6 active people and the 3 level of links?

Case 5:	**A Maximal path from node X to node Y**
Rumor mill	Find a Maximal path where rumor spread from person x to person y?
Disease network	How a disease spread from disease 1 to disease 2?
Cancer network	Find a frequent Maximal path from gene X to gene Y?
Facebook	Find a conversation Maximal path from person X to person Y?

Case 6:	**Finding the n^{th} frequent node and it connectivity**
Rumor mill	Find the 6^{th} trusted source and to whom all he passed the rumor?
Disease network	Find the 4^{th} contagious disease?
Cancer network	Find the 8^{th} frequent gene and its frequent interactions?
Facebook	Find the 3^{rd} active person on Facebook?

Case 7:	**Finding the most frequent node and its incoming, outgoing connections**
Rumor mill	Find the most trusted source, From whom he received and to whom he passed the rumor?
Disease networ	Find the most deadly disease, the cause and effects of the disease?
Cancer networ	Find the most frequent gene; find it incoming and outgoing interactions?
Facebook	Find the most active member on Facebook, find incoming and outgoing interaction?

Case 8:	**Finding the n^{th} frequent node and its incoming, outgoing connections**
Rumor mill	Find the 6^{th} trusted source, From whom he received and to whom he passed the rumor?
Disease network	Find the 4^{th} contagious disease, the cause and effects of the disease?
Cancer network	Find the 8^{th} frequent gene; find it incoming and outgoing interactions?
Facebook	Find the 3^{rd} active person on Facebook, find incoming and outgoing interaction?

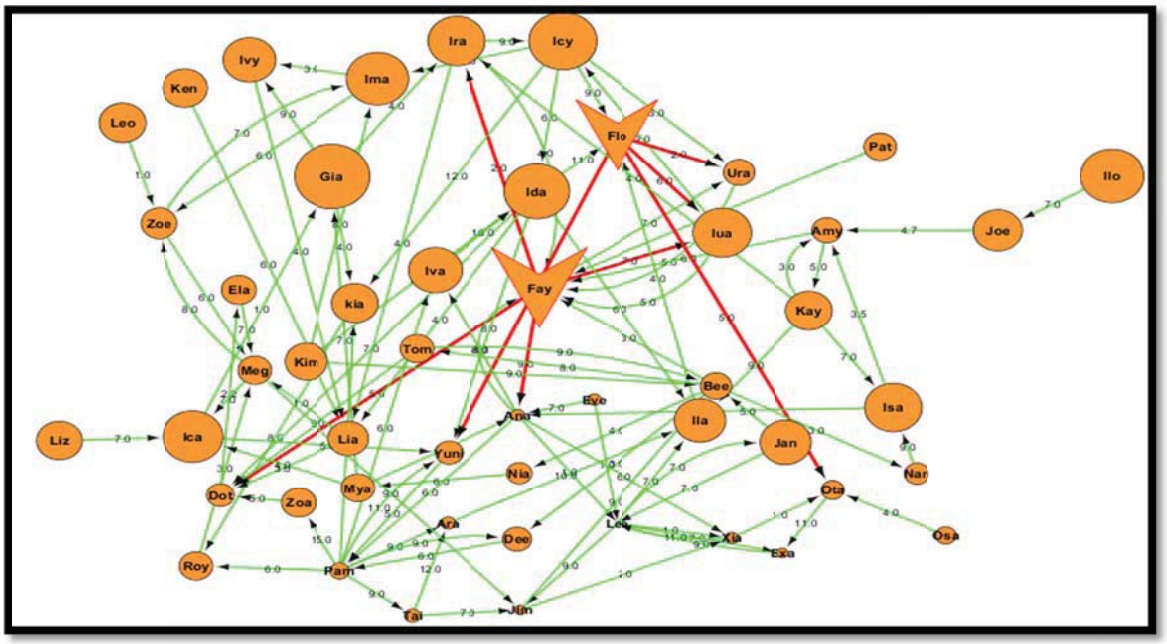


Figure 7: A subnetwork showing the most two famous people in the group and to whom all they communicate. Note: Visualization of Figure 5(a) in section 3.4.3.

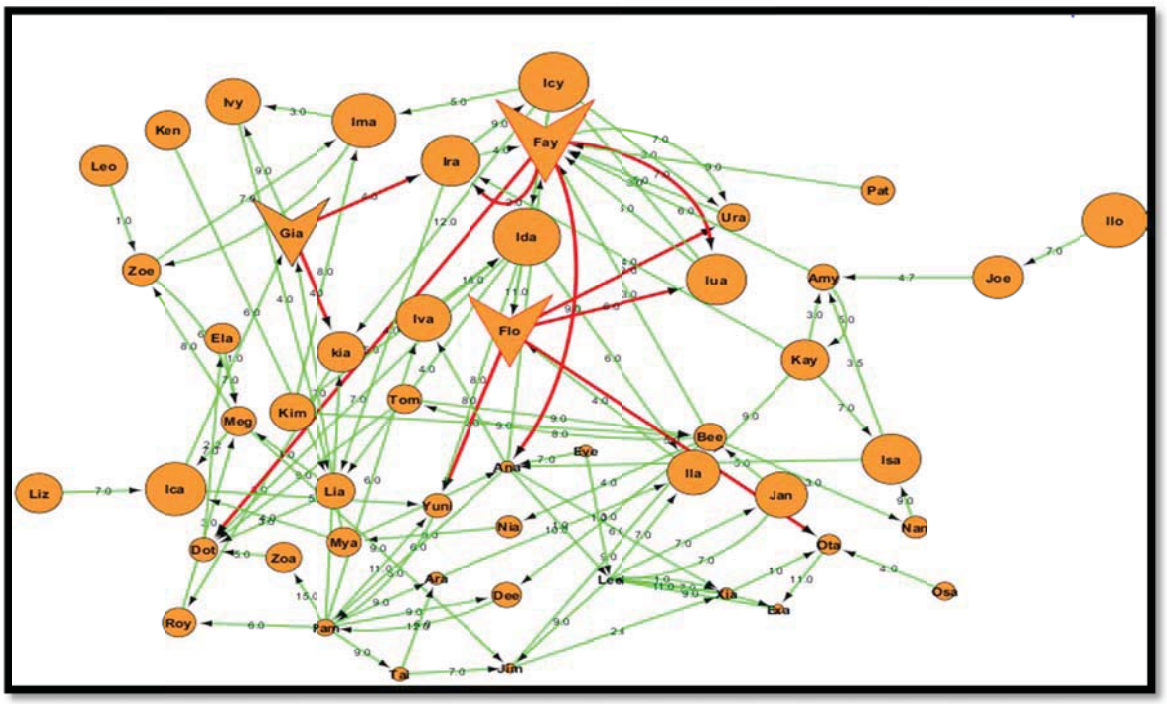


Figure 8: A subnetwork showing n number of famous people and the communication (where n= 3). Note: Visualization of Figure 5(d) in section 3.4.3.

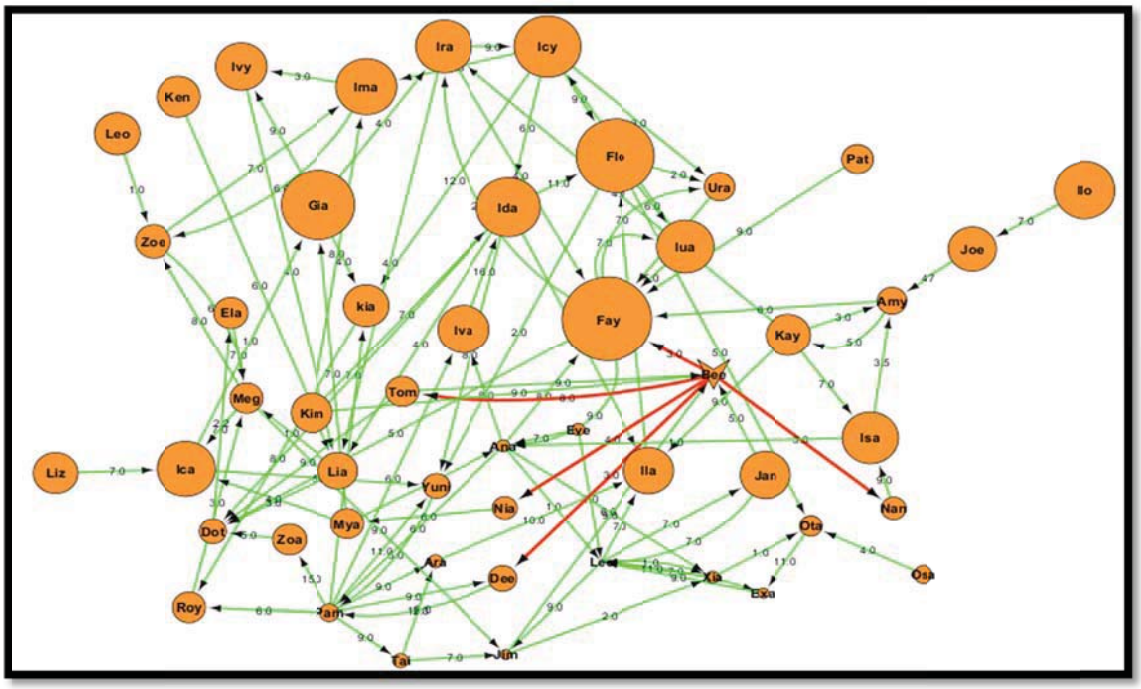


Figure 9: A subnetwork showing the nth famous person and to whom all they communicate (where n= 32). Note: Visualization of Figure 5(e) in section 3.4.3.

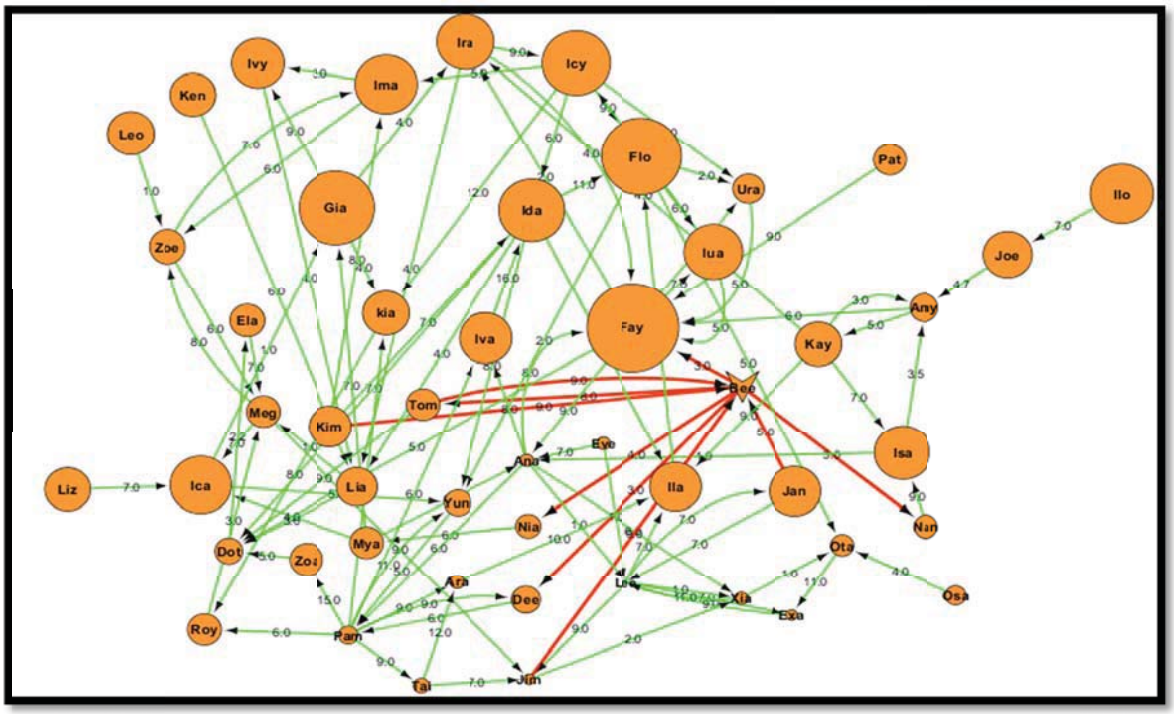


Figure 10: A subnetwork showing nth famous person and his/her incoming and outgoing communication (where n=32). Note: Visualization of Figure 5(h) in section 3.4.3.

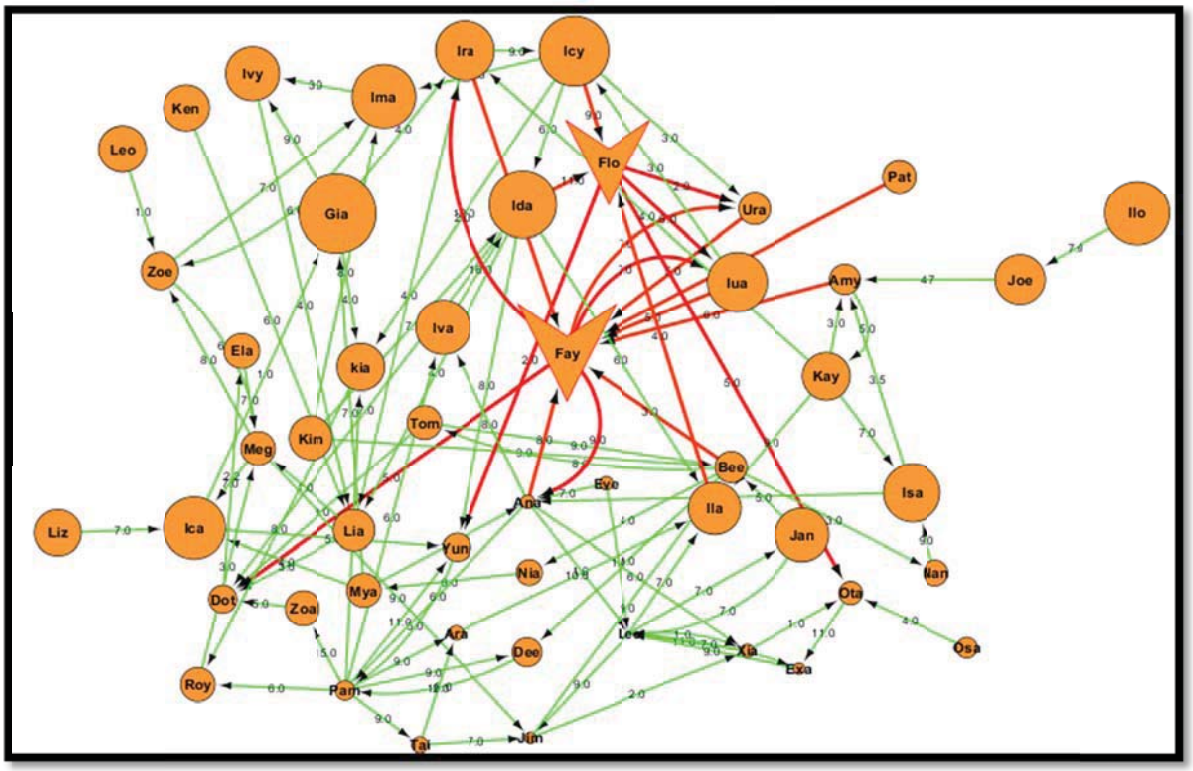


Figure 11: A subnetwork showing the most two famous people and their incoming and outgoing communication pattern. Note: Visualization of Figure 5(g) in section 3.4.3.

Figures 7, 8, 9, 10, 11 are the visualizations of 5 different cases of the algorithm from section 3.4.3. Node weight is represented using the size of the nodes (bigger the size of a node, higher the node weight). Edge weights are marked in the Figures (the numerical values). Directions are represented using arrows.

4.1.2 Rumor mill

Let us consider an example of rumor mill in order to demonstrate the result of case 5(f) (Possible highly weighted pathways in a network) from Figure 5 in section 3.4.3. Consider the exact network as in the example of Facebook but in this case the network was modeled as a Rumor Mill such that the nodes represent people and edge

represents the interaction. The nodes weight $W_{N(i)}$ represent the trusted source, the edges weight $W_{E(ij)}$ represent the amount of information (rumor) passed and the direction indicates the flow of information.

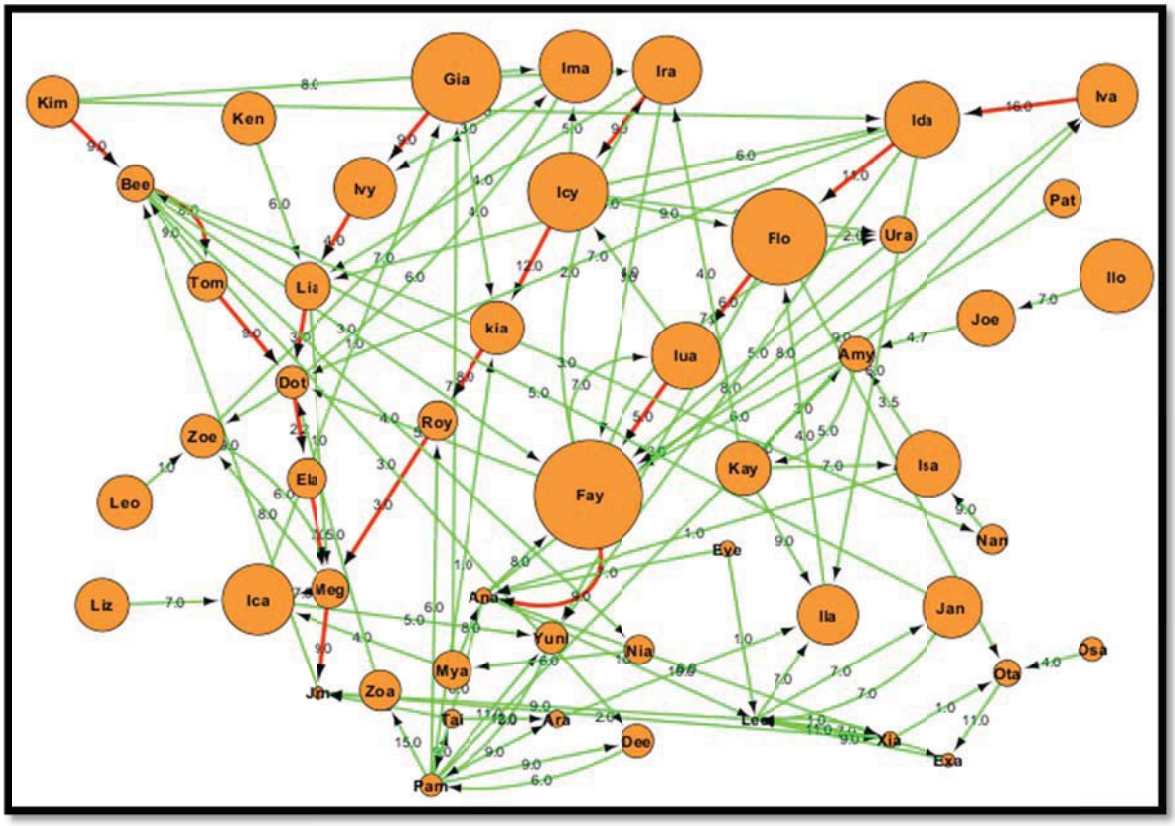


Figure 12: Some of the Maximal paths where the maximum rumor being passed among people. Note: Visualization of Figure 5(f) in section 3.4.3.

Figure 12 is a visualization of 4 different Maximal paths discovered as a result of the algorithm. A total of 10 Maximal paths were identified as a result of case 5(f) taking ρ=2. These Maximal paths were ranked based on the method illustrated in Section 3.5(iii). The Maximal path shows the pattern by which maximum rumor being passed between people. Table 2 displays the 9 Maximal paths discovered as a result of the algorithm.

Table 2: Maximal paths derived using the proposed algorithm and ranking. Note: If score = '0'. Then it is an invalid Maximal path and will not be considered for ranking.

No	Maximal paths	Score 1 at β=50%	Rank based on score 1	Score 2 at β=70%	Rank based on score 2
1	$Iva \rightarrow Ida \rightarrow Flo \rightarrow Iua \rightarrow Fay \rightarrow Ana$	254	1	208	1
2	$Kay \rightarrow Ila \rightarrow Flo \rightarrow Iua \rightarrow Fay \rightarrow Ana$	239	2	208	1
3	$Ira \rightarrow Icy \rightarrow Kia \rightarrow Roy \rightarrow Meg \rightarrow Jim$	89	6	49	4
4	$Gia \rightarrow Ivy \rightarrow Lia \rightarrow Dot \rightarrow Ela \rightarrow Meg$	93	5	56	3
5	$Ilo \rightarrow Joe \rightarrow Amy \rightarrow Fay \rightarrow Ana$	146	3	45	5
6	$Kim \rightarrow Bee \rightarrow Tom \rightarrow Dot \rightarrow Ela \rightarrow Meg$	31	7	0	–
7	$Nan \rightarrow Isa \rightarrow Amy \rightarrow Fay \rightarrow Ana$	99	4	0	–
8	$Ken \rightarrow Lia \rightarrow Dot \rightarrow Ela \rightarrow Meg \rightarrow Jim$	31	7	0	–
9	$Zoa \rightarrow Dot \rightarrow Ela \rightarrow Meg \rightarrow Jim$	0	_	0	–

Validation:

All the Maximal paths and the subnetwork derived were manually verified. A Maximal path derived gives the following information: how is the rumor passed? From where the rumor started and where did it end? and which is the Maximal path where maximum rumor passed from source to destination? For example, consider a Maximal path from Figure 2 $Iva \rightarrow Ida \rightarrow Flo \rightarrow Iua \rightarrow Fay \rightarrow Ana$. This Maximal path tells us that the rumor started from 'Iva' and ended at 'Ana'. And 'Iva' passed the maximum rumor to 'Ida' than anyone else. Similarly 'Ida' passed maximum rumor to 'Flo' and so on. The node weight can also tell us that this Maximal path has many trusted people compared to other paths, hence giving it a higher rank (See Figure 12).

4.2 Real time datasets

We used colorectal cancer network to evaluate the performance of the proposed method for mining significant Maximal paths from a biological network. The network

was chosen in such a way that the most important maximal paths and sub networks are already identified and well-known from the literature. The motivation of the exploration is to prove that the Maximal paths found by algorithm with the help of node weight, edge weight, and direction is biologically important and meaningful, as well. The ultimate goal of the algorithm is to find novel and biologically meaningful pathways. In this section, we illustrate a systematic performance study on the selected biological network data set. The experiments were conducted on a computer with a Pentium M 1.4 G CPU and 512 M main memory, running Microsoft Windows XP operating system. We implemented the algorithm in C.

4.2.1 Biological dataset 1 (Apoptosis colorectal cancer)

We used a list of 45 protein entities associated with colon cancer and text mining was done using our in-house tool BioMAP [85] to discover protein-protein associations. The protein entities were uploaded to the Metacore^{TM} and were analyzed for their pathways and functions. Many pathways were generated using Metacore^{TM}. For this paper we considered only the first six pathways associated with Apoptosis. Both genes and the interactions between these genes were extracted manually and their directionality was obtained as well. There were a total of 150 genes and 290 directional interactions in the six pathways of Apoptosis. A new network called the main network was modeled using the 150 genes, the interaction between these genes and direction using Cytoscape.

Network modeling:

The next step was to incorporate the node/genes and edge/interaction parameters into the main network which is the node weight and the edge weight. Degree of each

node was calculated using the Cytohubba [90] function in Cytoscape. This degree was considered as the node weight in our experiment. The gene ontology distance [7] was calculated between every interaction in the main network using a graph based formula [110], thus the gene ontology distance was obtained for the 290 directional interactions. This gene ontology distance was considered as the edge weight, thus modeling a weighted, directed network. We now have a network for Apoptosis which has node weight as degree and edge weight as gene ontology distance, along with directional edges.

Analysis:

We followed the procedure listed under methodology to form the canonical matrix. For this particular experiment, the threshold σ was set at 2, which means the nodes with degrees less than 2 were eliminated. The subnetwork mining algorithms were carried out specifically looking for interesting paths.

The Apoptosis network was mined using the proposed algorithm and 120 Maximal paths were discovered. This was cross verified with Metacore^{TM} where 10% of the Maximal paths were identically matched as compared to the listed pathways in Metacore^{TM}. The rest of the Maximal paths had few edges missing or few extra edges inserted compared to Metacore^{TM}. Those extra edges were cross verified with literature and were proven to be significant. It was found that Metacore^{TM} does not have all maximal paths and genes as it lacked in identifying some of the genes which appeared

Table 3: Maximal paths derived and scoring. Note: Arrows and letter 'X' in red shows the missing edge and node respectively. Accuracy rate is the comparison of each pathway with a similar pathway found from literature.

No.	Maximal Paths derived at σ = '2', Node weight = "degree of nodes", Edge weight = "Gene Ontology distance".	Path Score.	Accuracy	
			Accuracy Rate	Reference from literature
1.	TNF-alpha→ TNF-R1→TRADD→ FADD→ Caspase-8 → Caspase-10	9.0167	1	[27]
			0.83	[116]
2.	FasL→ FasR → FADD→ Caspase-8 → Caspase-10	9.01006	1	[116]
			0.8	[27]
3.	TWEAK→ DR3→ TRADD→ FADD→ Caspase-8→ Caspase-10	8.5519	0.75	[108]
			0.86	[27] for Casp8-Casp10
4.	Cytochrome c→ APAF-1 → Caspase 9→ Caspase-7	7.6478	1	[26]
			0.75	[56]
5.	TNF-alpha→TNF-R1→TRADD→ RIPK1→ IKKGAMMA→ IKKCAT→I-kB→NF-kB→c-IAP2	6.28154	0.77	[163]
6.	Apo-2L→DR5→TRADD→ 'X' →MEKK1 → MKK7→ JNK→ Bcl-2	5.3798	0.75	[108]

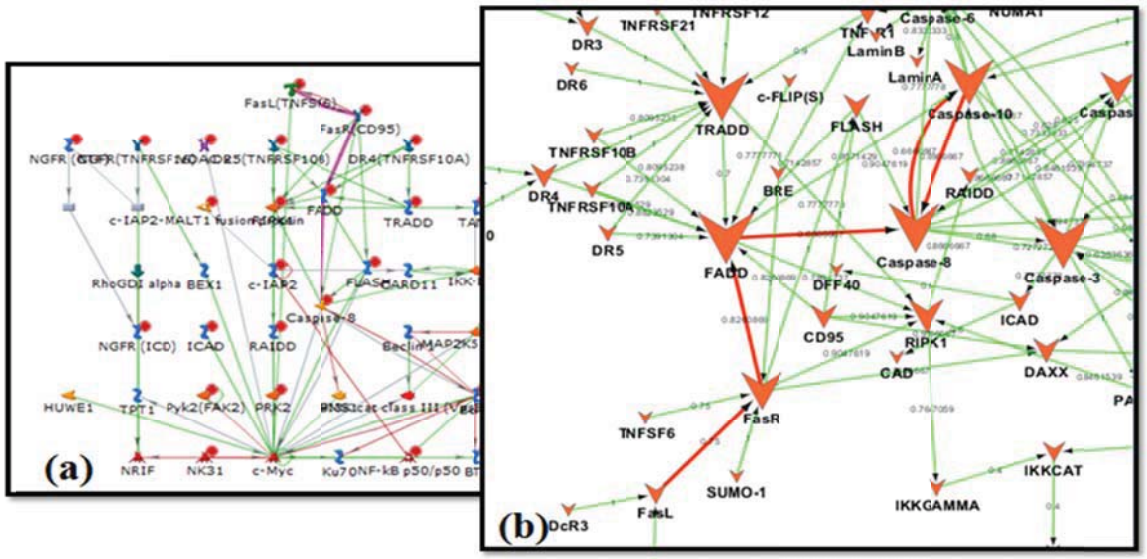


Figure 13: Maximal path validation 1. Note: (a). is a Metacore™ pathway Fasl-FAsr- Caspase 8 [23]. (b). A Maximal paths found using the proposed algorithm. Network (b) is modeled with node weight, edge weight and direction. The size of the node significance is the node weight (degree). The edges are marked with the edge weight (gene ontology distance). As seen in (b) the Maximal paths starts from Fasl not from DcR3 because of the fact that σ= 2.

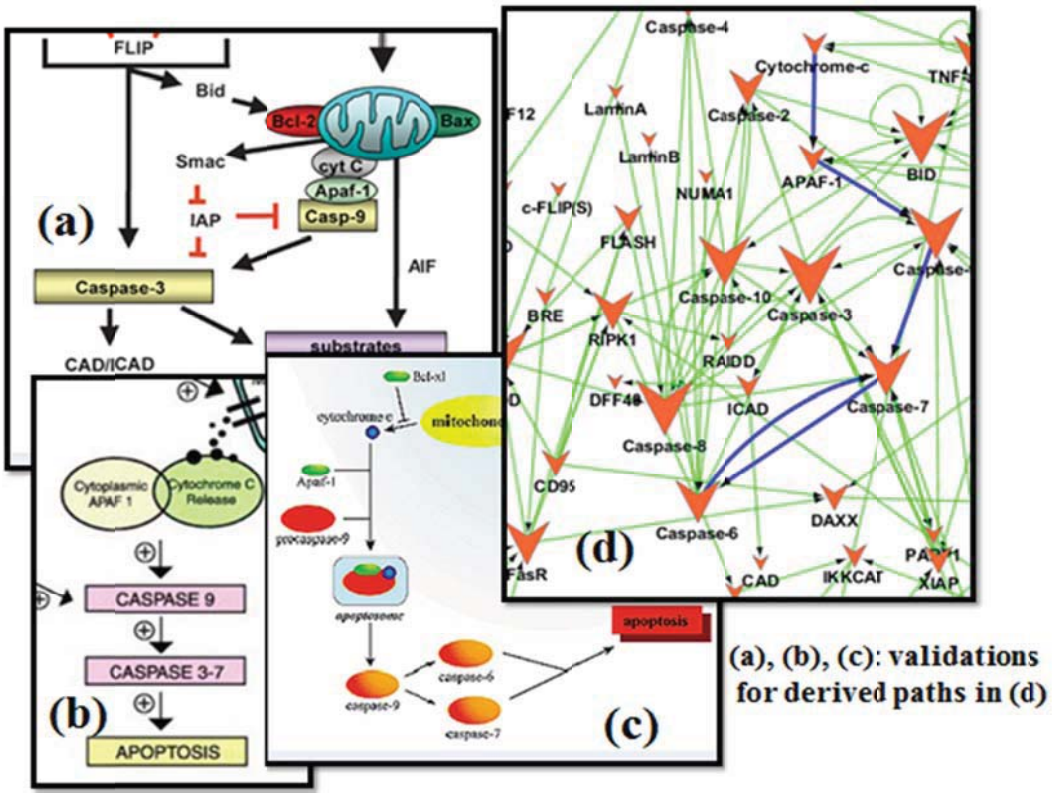


Figure 14: Maximal path validation 2. Note: (a),(b),(c). are validations for paths "Cyt C → Apaf-1→ Casp-9 → Casp-7 → Casp-6" from literature review [56], [26], [153] respectively. 5(d) shows a pathway found using the proposed algorithm. Network (d) modeled with the node weight, edge weight and direction. The size of the node significance is the node weight (degree). The edge weights are gene ontology distance (not displayed).

in the Apoptosis network. Hence, this algorithm is an efficient way of finding meaningful biological Maximal paths based on node weight, edge weight, and directionality. Also, the results prove that Gene Ontology distance and degree are two parameters that can be well used to find interesting Maximal paths from a biological network.

Table 3 shows some of the Maximal paths discovered as a result of the algorithm. A total of 90 unique Maximal paths were discovered. Those listed are just a few pathways selected randomly from the 90 paths. The six pathways listed were crossed verified with Maximal paths from literature and an accuracy rate was calculated.

Figure 13 and Figure 14 illustrate two examples of the significance of the Maximal paths that were discovered. Figure 14(a) is a network from Metacore™ with a functional pathway called as "Fasl-Fasr-Caspase-8" as marked. Figure 14(b) shows a path1: Fasl-Fasr-fadd-Caspase8-Caspase 10-caspase-8. This Maximal path was discovered using the edge weight and node weight. Thus, it is proven that this Maximal path is a functional pathway listed in Metacore™ along with an additional gene Caspase-10. The relation between Caspase-8 and Caspase-10 was verified from literature. Hence, we can prove that this is a significant Maximal path that was discovered.

Figure 14 shows the literature validation of the pathway derived by the proposed algorithm. Figure 14(d) is the derived pathway $CytC \rightarrow Apaf1 \rightarrow Casp9 \rightarrow Casp7 \rightarrow Casp6$. As seen in Figure 14(a) Casp-9 interacts with Casp-3 following to Apoptosis which is different from the derived pathway. Even though 14(a) is a very well-known pathway, there exists other validations such as 14(b) and 14(c) showing the interaction of 'Casp-9' to 'Casp-7' and then to 'Casp-6'. Hence, this is a strong literature validation proving that the pathway derived using the proposed algorithm is correct and significant.

Figure 15 shows the literature validation of the pathway derived by the proposed algorithm. Figure 15(b) is the derived Maximal path $TNF - alpha \rightarrow TNF - R1 \rightarrow TRADD \rightarrow RIPK1 \rightarrow IKKGAMMA \rightarrow IKKCAT$. Figure 15(a) is a literature validation from Metacore™.

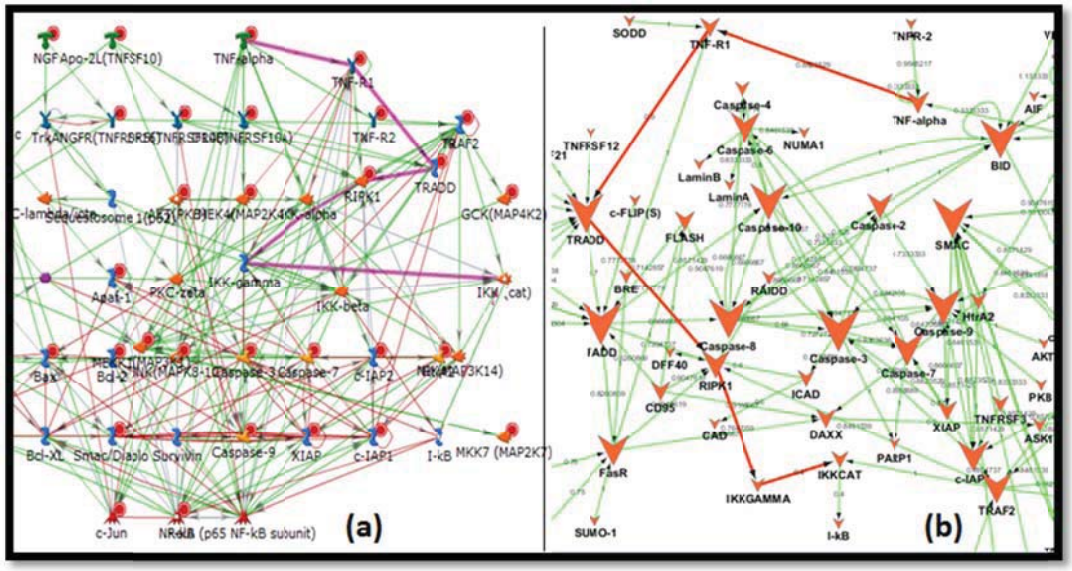


Figure 15: Maximal path validation 3. Note: (a). is a Metacore^{TM} pathway TNF-alpha-TNF-R1- IKK [108]. (b). A Maximal paths found using the proposed algorithm. Network (b) is modeled with node weight, edge weight and direction. The size of the node significance is the node weight (degree). The edges are marked with the edge weight (gene ontology distance). As seen in (b) the Maximal paths ends at IKKCAT not at I-kB because of the fact that σ= 2.

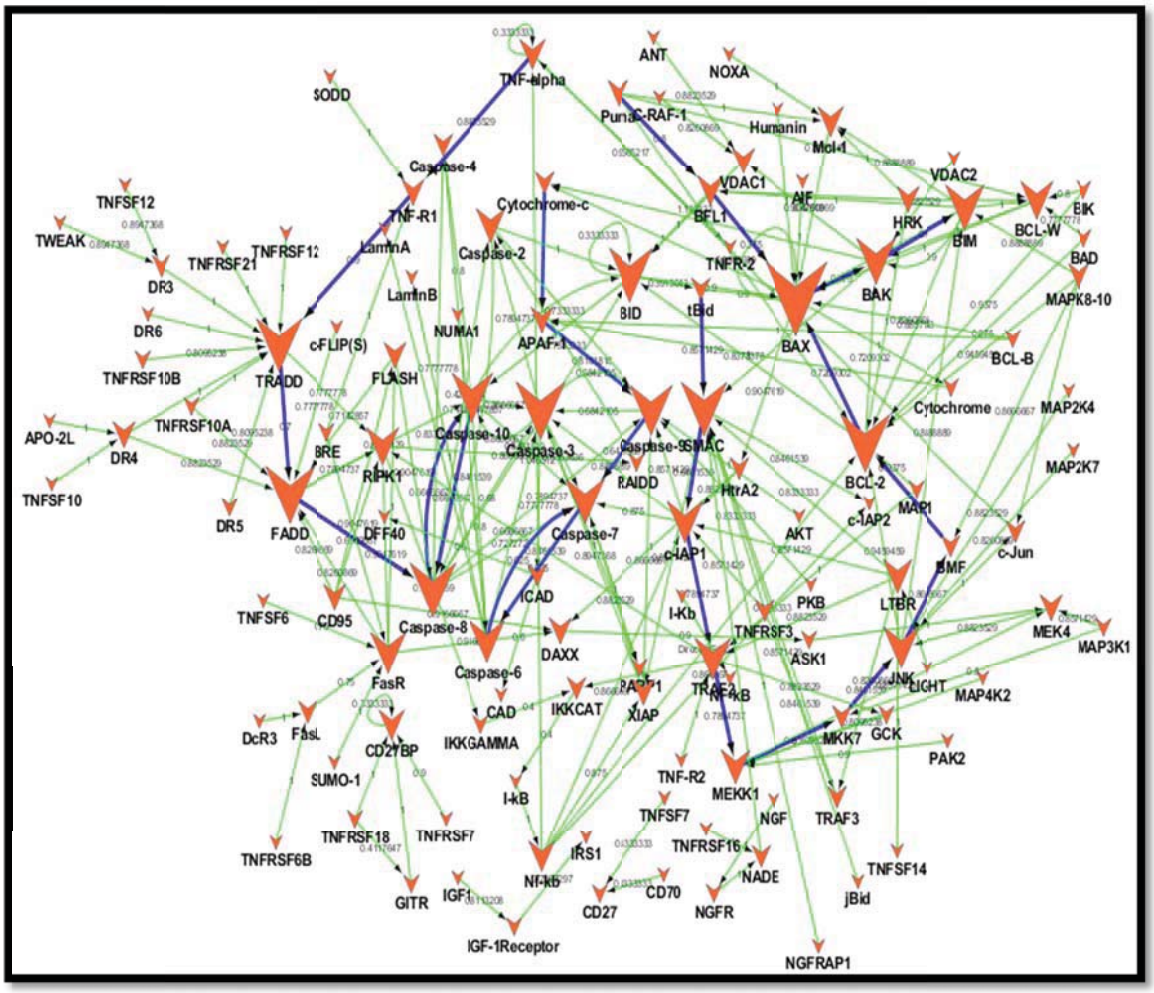


Figure 16: An apoptosis network. Note: It is having a total of 150 genes and 290 directional interactions. The size of the node signifies the degree (node weight) and edges are marked with gene ontology distance (edge weight). Four Maximal paths are displayed in the above network out of the 90 Maximal paths discovered using the proposed algorithm.

Figure 16 is the entire apoptosis network with 150 genes and 290 interactions between them. The Maximal paths marked in the network are few Maximal paths derived as a result of the algorithm.

The results prove that the proposed algorithm is an efficient way to find significant Maximal paths from a biological network given the appropriate edge weight, node weight and direction. Using the proposed algorithm we can also find the right parameters to

predict the interaction Maximal paths in a network. From the above performed experiments it is proved that the gene ontology distance is an appropriate parameter to find the Maximal paths from a network.

Validation:

The Maximal paths discovered were validated using Metacore™ and it was found that 72% of the Maximal paths discovered using the algorithm was among the functional pathways of Metacore™. The Maximal paths were also cross verified with literature.

4.2.2 Biological dataset 2 (Colorectal cancer)

We used a list of protein entities associated with colon cancer and text mining was done using our in-house tool BioMAP [85] to discover protein-protein associations. The protein entities were uploaded to the Metacore™ and were analyzed for their pathways and functions. In Metacore™ the gene content of the uploaded files is used as the input list for generation of biological networks using Analyze network algorithm with default settings. This is a variant of the shortest Maximal paths algorithm. These networks are built on the fly and unique for the uploaded data. In this workflow the networks are prioritized based on the number of fragments of canonical pathways on the network. From the network list we selected the highest priority list as shown in table 4. But it was found that many genes from the list were missing in the network created by Metacore™. This was how the list of genes selected to model a network.

Table 4: Metcore^{TM} network.

No	Key network objects	GO Processes	Total nodes	Root nodes	Pathways	p-Value	zScore	gScore
1	SP1, CBP, c-Myc, STAT5A, SMAD1	multicellular organismal reproductive process (45.8%), multicellular organism reproduction (45.8%), reproductive process (52.1%), reproduction (52.1%), organ morphogenesis (37.5%)	50	15	49	2.39e-38	87.07	148.32

Network Modeling:

The network for the experiment was modeled with degree as the node weight, gene ontology distance as the edge weight and the directions were relevant from the protein interaction. The input dataset was a small dataset with 36 nodes (genes/protein) and 126 edges (interactions). The result of the algorithm was cross verified with the pathways in Metacore$^{TM.}$ Eighteen Maximal paths were discovered as a result of case 5(f) from Figure 5 in section 3.4.3. Figure 17 and Figure 18 shows two such discovered paths. Figure 19 are the subnetworks derived as a result of the algorithm. Table 5 shows some of the Maximal paths derived as a result of the algorithm and the Maximal paths are scored based on the method discussed in section 3.5(i).

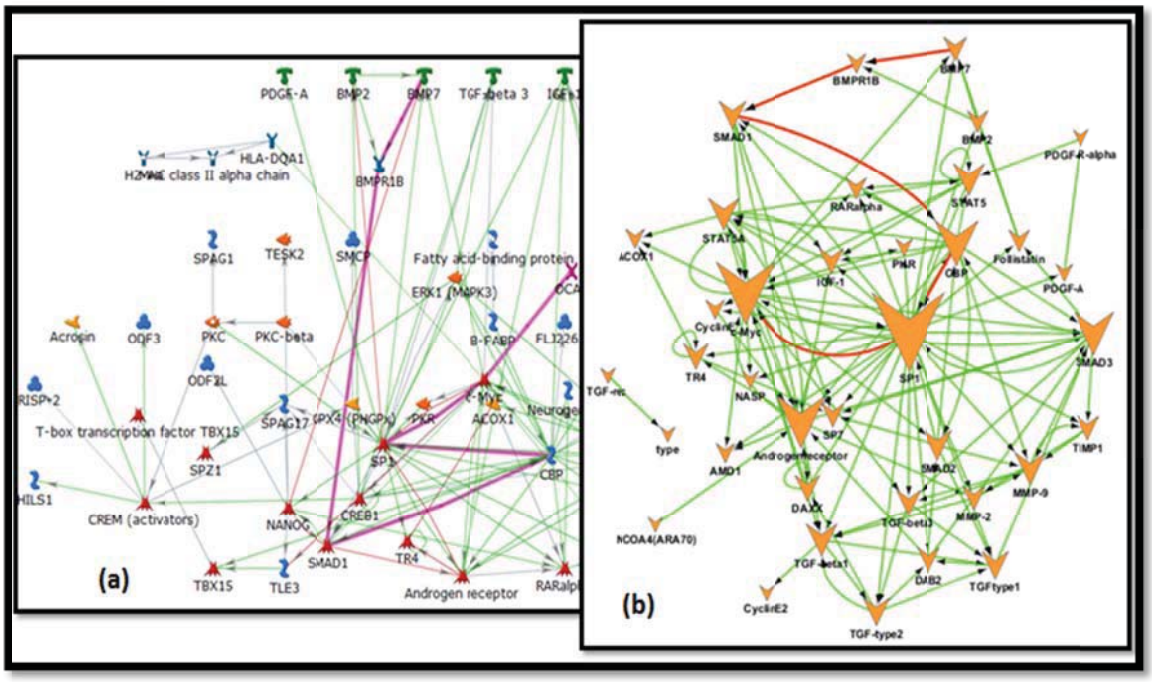


Figure 17: Maximal path validation 1. Note: (a). is a Metacore™ pathway BMP7-BMPR1 –OCA3 [108]. (b). A Maximal path found using the proposed algorithm. Network (b) is modeled with node weight, edge weight and direction. The size of the node significance is the node weight (degree).

Table 5: Few Maximal paths derived as a result of the algorithm and scoring.

No.	Maximal Paths derived at σ = '2', Node weight = "degree of nodes", Edge weight = "Gene Ontology distance".	Path Score
1	$PDGF-A \rightarrow PDGF-R-ALPHA \rightarrow STAT5 \rightarrow SP1 \rightarrow MMP-2 \rightarrow TGF-BETA1 \rightarrow CYCLINE2$	8.3846
2	$STAT5A \rightarrow C-MYC \rightarrow TGF-BETA3 \rightarrow TGF-TYPE2 \rightarrow TGF-TYPE1$	9.444
3	$TGFTYPE1 \rightarrow SMAD3 \rightarrow C-MYC \rightarrow BMP7 \rightarrow BMPR1B \rightarrow SMAD1 \rightarrow CBP$	9.3846
4	$BMPR1B \rightarrow CBP \rightarrow C-MYC \rightarrow SP1$	14.42
5	$CYCLINE2 \rightarrow ANDROGENRECEPTOR \rightarrow TR4 \rightarrow SP7$	8.28

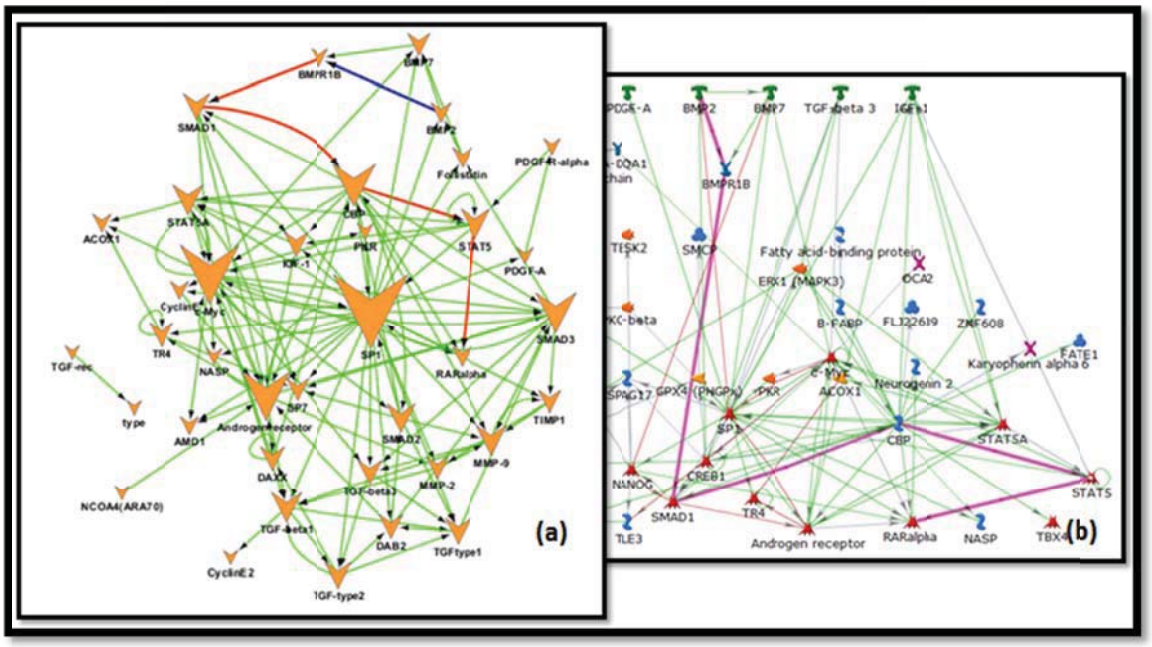


Figure 18: Maximal path validation 2. Note: (a). A Maximal path found using the proposed algorithm. Network (b) is modeled with node weight, edge weight and direction. The size of the node significance is the node weight (degree). The edge highlighted in blue was a missing edge in the derived Maximal path compared to Metacore™ pathway (b) is a Metacore™ pathway BMP2- BMPR1 –RARalpha. [108]

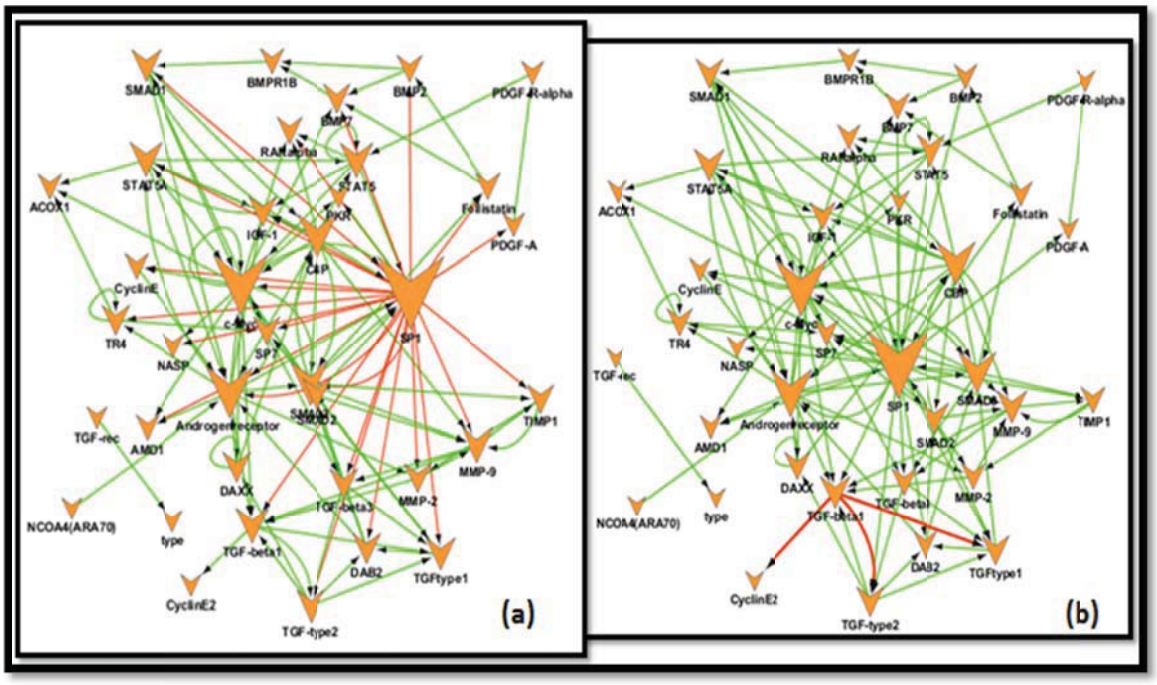


Figure 19: Subnetwork. Note: (a). A subnetwork showing the most highly weighted gene and it's outgoing interaction (A hub node). Visualization of Figure 5(a) in section 3.4.3 (b). A subnetwork displaying the 9th highly weighted gene in the network and it outgoing interactions. Visualization of Figure 5(e) in section 3.4.3 (b).

Validation:

The Maximal paths discovered as a result of case 5(f) from section 3.4.3 were validated using Metacore^{TM}. Other results were cross verified manually since the network size was very small.

4.2.3 Biological dataset 3 (Colorectal cancer in three domains)

We used a set of experiment list and also mined the literature to get colorectal cancer data. With the help of BioMAP we augmented the experimental data with the literature data to find significant genes associated with colorectal cancer. The associations between the genes were identified using an in-house PPI algorithm (Protein Protein Interaction). The extraction resulted in 576 genes and 1424 interactions.

Then gene ontology was computed for every gene product. The gene ontology provides ontology of defined terms representing gene product properties. The ontology covers three domains: Cellular component, the parts of a cell or its extra cellular environment; molecular function, the elemental activities of a gene product at the molecular level, such as binding or catalysis; and biological process, operations or sets of molecular events with a defined beginning and end, pertinent to the functioning of integrated living units: cells, tissues, organs and organisms. [7]

The Gene ontology was calculated based on three domains: cellular component, molecular function and biological process. The gene ontology distance was calculated for all the three domains separately using the Graph structure based formulae. [110]

Network modeling:

Since three different domains (cellular component, molecular functions and biological process) were considered, there were three different networks for the experimental study: a network formed by the cellular component, a network formed by molecular component, a network by biological process. The next step was to incorporate the nodes/genes and edge/interaction parameters into the network which is the node weight and the edge weight. Degree of each node was calculated using the Cytohubba [90] function in Cytoscape. This degree was considered as the node weight in our experiment. The gene ontology distance [7] was calculated between every interaction in the network using a graph structure based formula [110], thus the gene ontology distance was obtained for the 1424 directional interactions. This gene ontology distance was considered as the edge weight, thus modeling a weighted, and a directed network. We

now have three networks on colorectal cancer, which has node weight as degree and edge weight as gene ontology distance, along with directional edges.

4.2.3.1 Network 1: (Domain1: Cellular component)

A cellular component is just that, a component of a cell, but with the provision that it is part of some larger object; this may be an anatomical structure (e.g. rough endoplasmic reticulum.

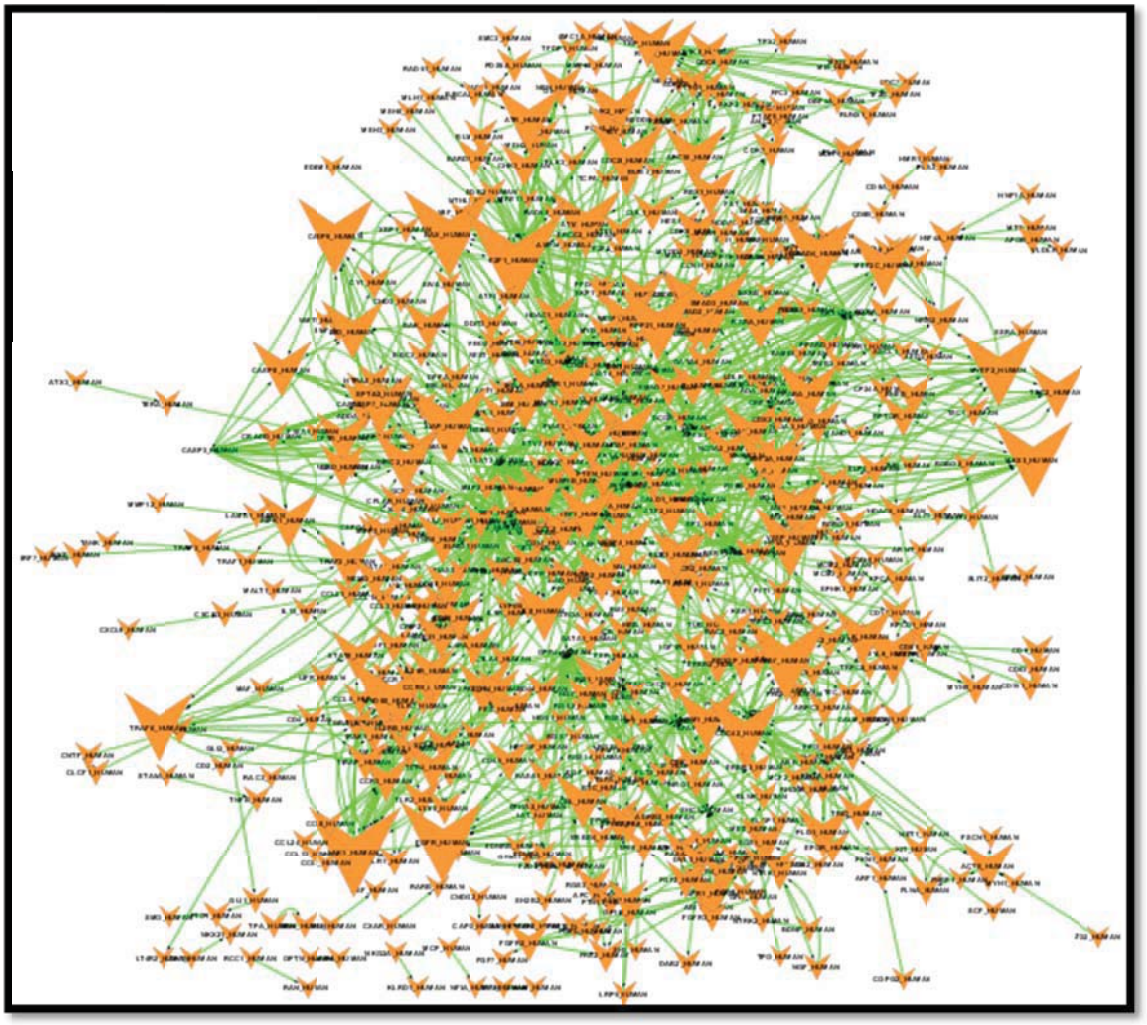


Figure 20: A colorectal cancer network with 1424 association between 576 genes.

or nucleus) or a gene product group (e.g. ribosome, proteasome or a protein dimer). Figure 21 shows the visualizations of the network. The network in cellular Component domain had a size of 576 nodes and 1424 edges. Edges with edge weights =1 were eliminated since, gene ontology distance =1 signifies “no interaction between genes”. Hence the network size was reduces to 327 nodes and 471 edges.

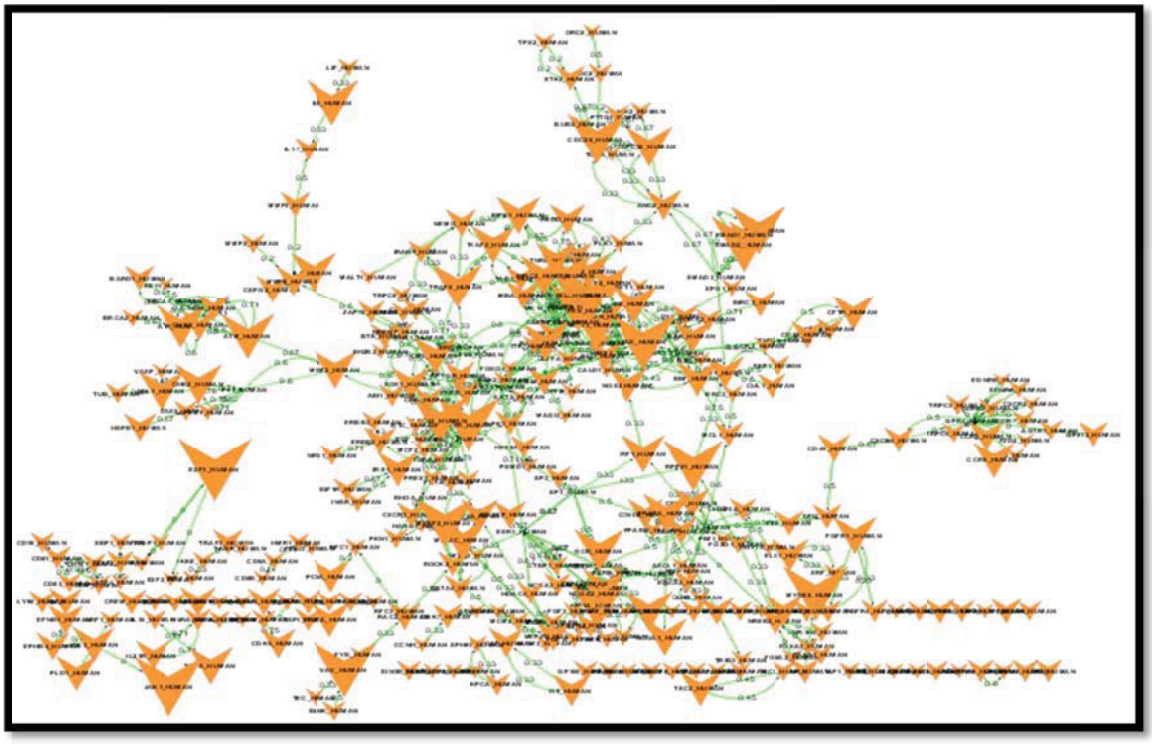


Figure 21: The input network to the algorithm.

Analysis:

We followed the procedure listed under methodology to form the canonical matrix. For this particular experiment, the threshold ‘σ’ was set at 2 which means the nodes with degrees less than 2 were eliminated. The subnetwork mining algorithms were carried out specifically looking for interesting Maximal paths and subnetworks.

The Colorectal cancer network under the cellular compound domain was mined using the proposed algorithm resulting in 34 Maximal paths and 28 subnetworks. Figure 22 shows three such discovered paths.

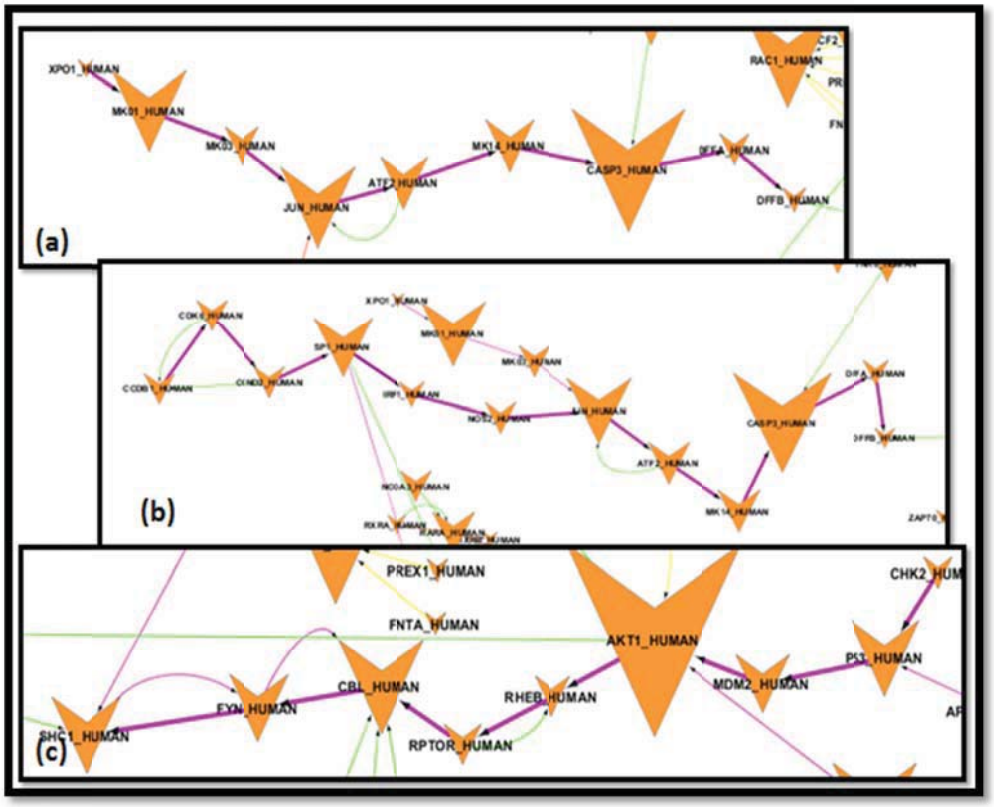


Figure 22: Some of the Maximal paths discovered as a result of the algorithm.

This was cross verified with Metacore^{TM} where 20% of the Maximal paths were identically matched compared to the listed pathways in Metacore^{TM}. The rest of the Maximal paths had few edges missing or few extra edges inserted compared to Metacore^{TM}. Those extra edges were cross verified with literature and were proven to be significant. It was found that Metacore^{TM} does not have all Maximal paths and genes as it lacked in identifying some of the genes which appeared in the colorectal cancer network. Hence, this algorithm is an efficient way of finding meaningful biological Maximal paths

based on node weight, edge weight, and directionality. Figure 23 shows some of the subnetworks derived as a result of the algorithm.

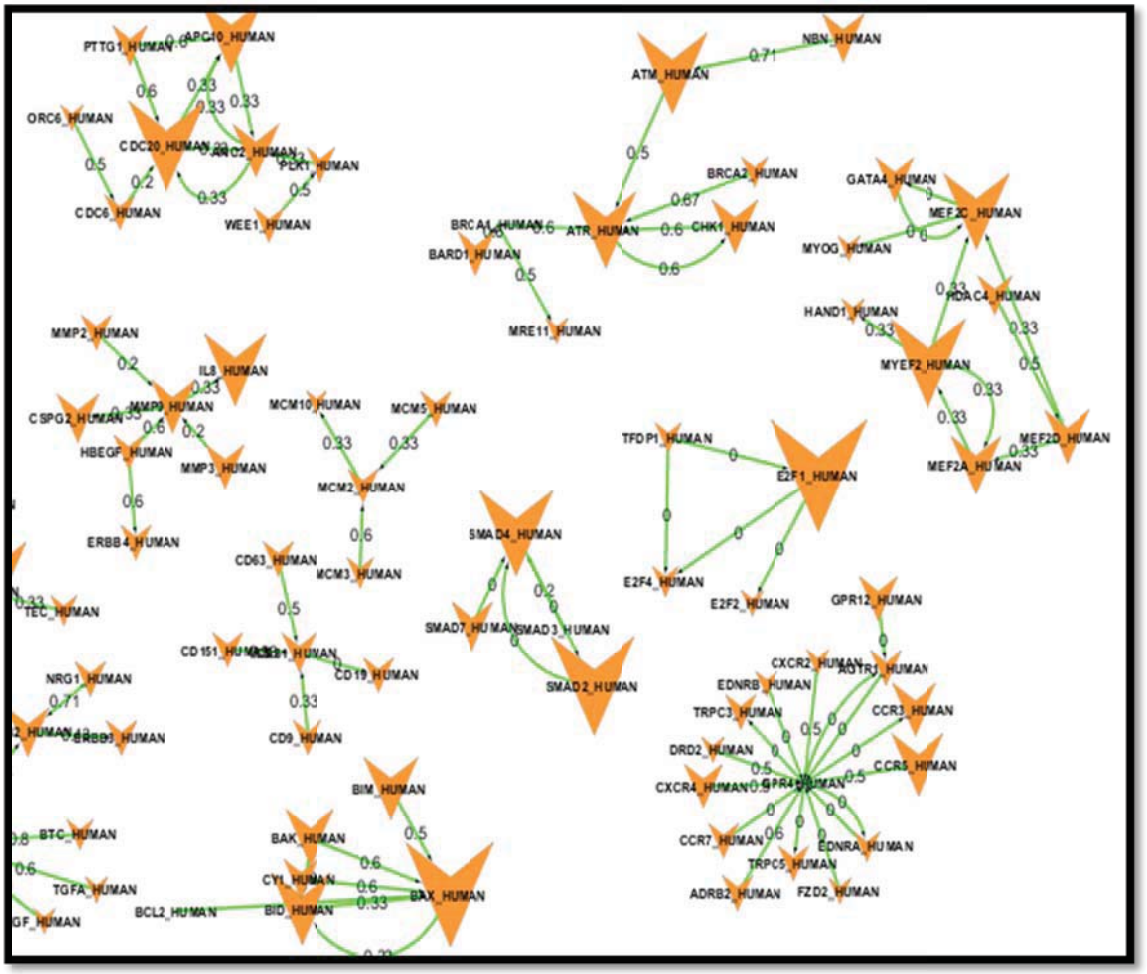


Figure 23: Subnetworks derived using the proposed algorithm.

Figure 24 and 25 shows the literature validation of generated subnetworks. The genes form the subnetwork was loaded into Metacore™ to form functional networks associated with the genes. The networks generated using Metacore™ was similar to the subnetworks generated as a result of the algorithm as shown in Figures.

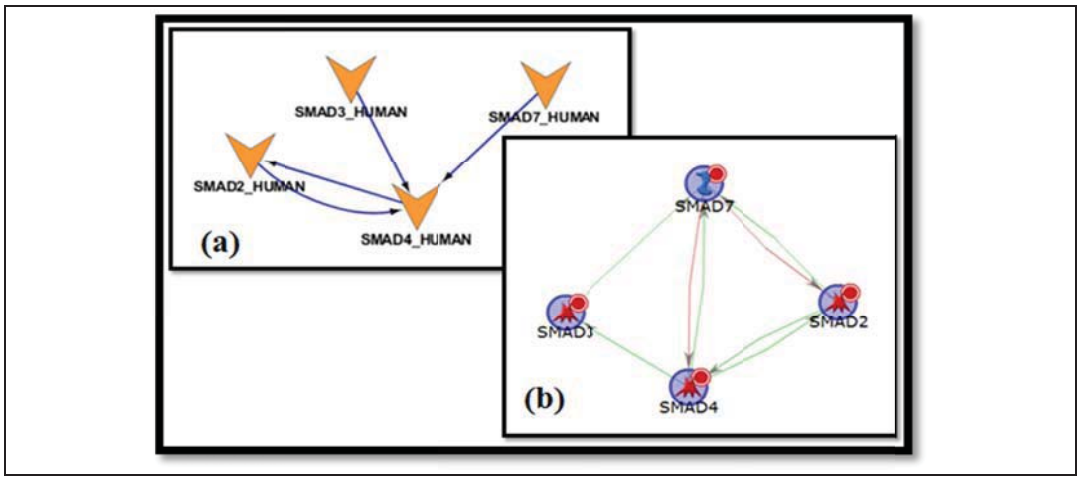


Figure 24: Subnetwork validation 1. Note: (a). A subnetwork derived using the proposed algorithm. Network (a) is modeled with node weight, edge weight, and direction. The subnetwork is derived with 'σ'=2. (b). A Metacore™ network generated as result of loading the list with genes SMAD3, SMAD4, SMAD2, SMAD7.

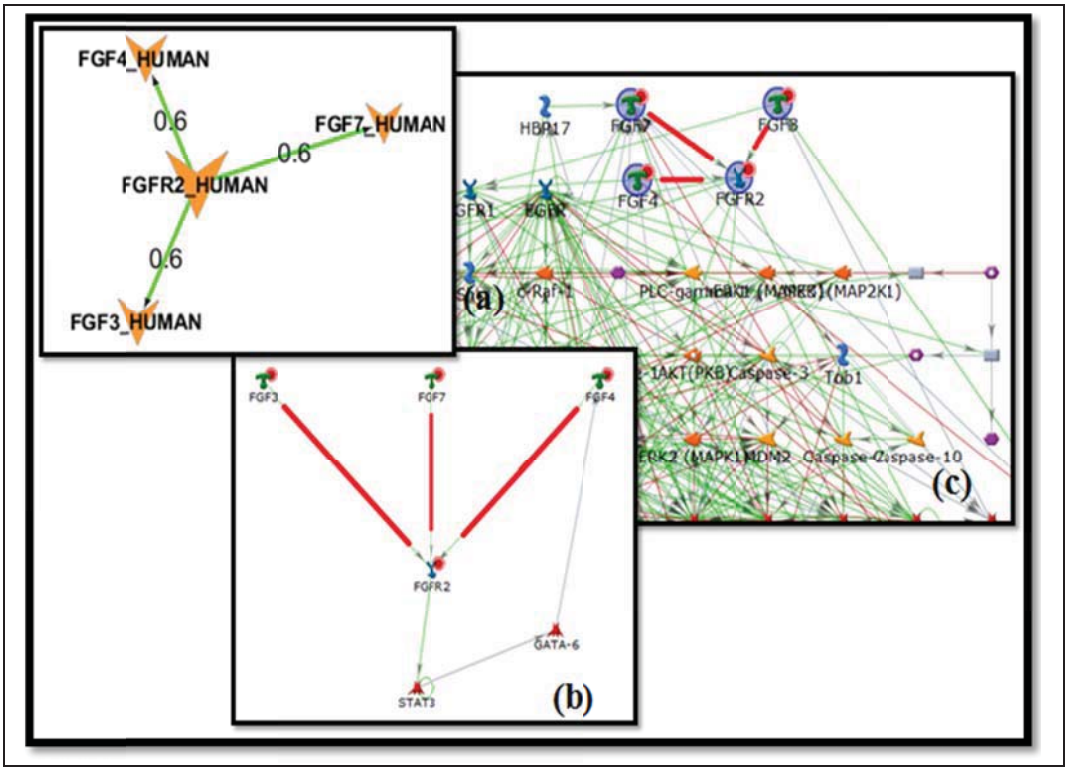


Figure 25: Subnetwork validation 2. Note: (a). A subnetwork derived using the proposed algorithm. Network (a) is modeled with node weight, edge weight, and direction. The subnetwork is derived with 'σ'=2. (b) and (c). Metacore™ networks generated as result of loading the list with genes from (a).

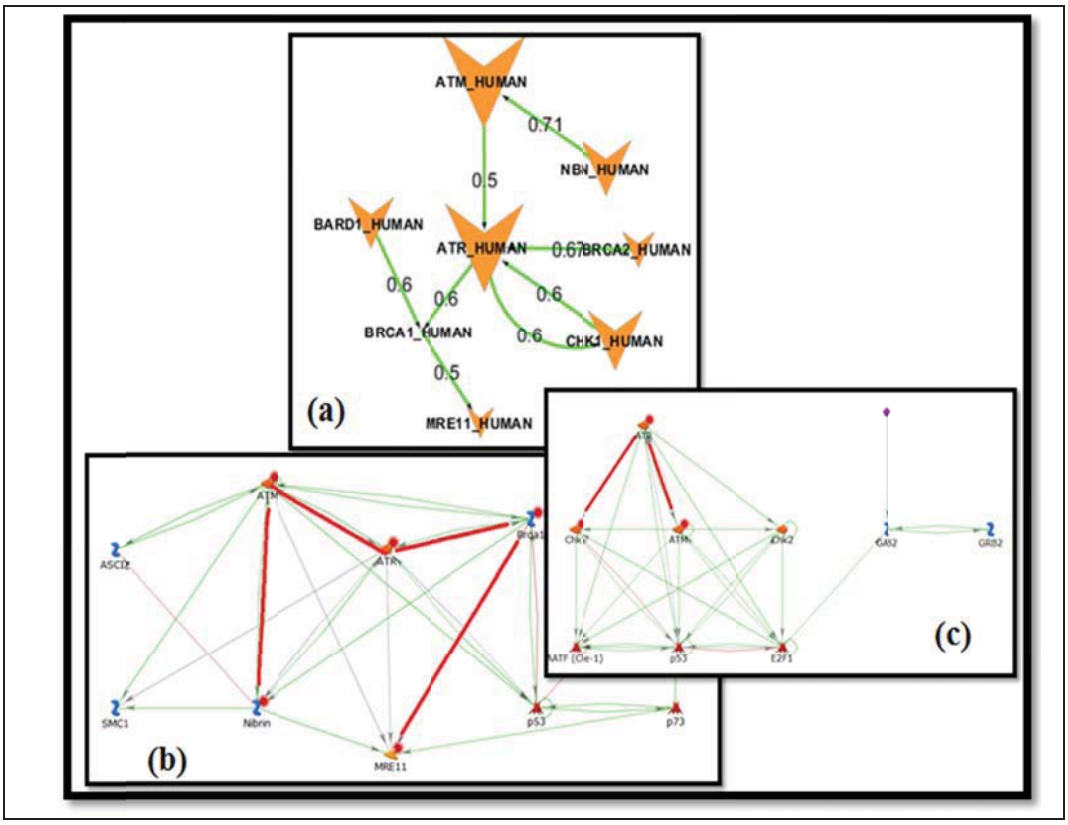


Figure 26: Subnetwork validation 3. Note: (a). A subnetwork derived using the proposed algorithm. Network (a) is modeled with node weight, edge weight and direction. The subnetwork is derived with 'σ'=2. (b) and (c). Metacore™ networks generated as result of loading the list with genes from (a).

Table 6: Maximal paths derived as a result of the algorithm, Maximal path scoring and ranking at β=40%.

No.	Maximal Paths derived at σ = '2', Node weight = "degree of nodes", Edge weight = "Gene Ontology distance".	Path Score	Score 1 at β=40 %	Rank based on score1
1	CHK2→P53→MDM2→AKT1→PDPK1	5.117	20	4
2	CCDB1→CDK6→CCND2→SP1→IRF1→NOS2→JUN→ATF2→MK 12→CASP3→DFFA--.DFFB	3.348	35	2
3	CCDB1→CDK6→CCND2→SP1→RPP21→STAT3→NMI	3.706	21	3
4	XPO1→MKO1→MKO3→JUN→ATF2→MK14→CASP3 →DFFA→DFFB	3.841	36	4
5	ASCL1→PPARA→NCOA1→RARA→RXRA	2.9266	0	-

Figure 26 is another example for subnetwork validation. In 26(b) it can be seen that the 4 out of 8 edges were exactly matching with 26(a). Similarly 26(c) is another network with similar edges and nodes. Network 26(b) and 26(c) were generated as a result of loading genes from 26(a) into Metacore™. Some of the genes are missing in the Metacore™ networks due to the lack of Metacore™ to identify those genes from the list.

Table 6 shows the Maximal paths derived as a result of the algorithm. 'Path score' is calculated using the method discussed in section 3.5(i). Score 1 is calculated using the method discussed in section 3.5(iii) at 'β=40%' (randomly assigned).

Validation:

The Maximal paths and the subnetworks discovered as a result of case 5(f) from section 3.4.3 were validated using Metacore™.

4.2.3.2 Network 2: (Domain 2: Molecular function)

A Molecular function describes activities, such as catalytic or binding activities, that occur at the molecular level. GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the action, and do not specify where or when, or in what context, the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene product, but some activities are performed by assembled complexes of gene products. The network in Molecular function domain has a size of 576 nodes and 1424 edges. Edges with edge weights =1 was eliminated since gene ontology distance =1 signifies "no interaction between genes". Hence the network size was reduces to 324 nodes and 568 edges.

Analysis:

We followed the procedure listed under methodology to form the canonical matrix. For this particular experiment, the threshold ‘σ’ was set at 2 which means the nodes with degrees less than 2 were eliminated. The subnetwork mining algorithms were carried out specifically looking for interesting Maximal paths and subnetworks. The Colorectal cancer network under the molecular function domain was mined using the proposed algorithm and 35 Maximal paths and 43 subnetworks were discovered. The Maximal paths and subnetworks discovered were entirely different compared to that of the cellular component domain. Figure 27 shows two such discovered paths. Table 7 shows the Maximal paths derived as a result of the algorithm. ‘Path score’ is calculated using the method discussed in section 3.5(i). Score 1 is calculated using the method discussed in section 3.5(iii) at ‘β=25%’ (randomly assigned).

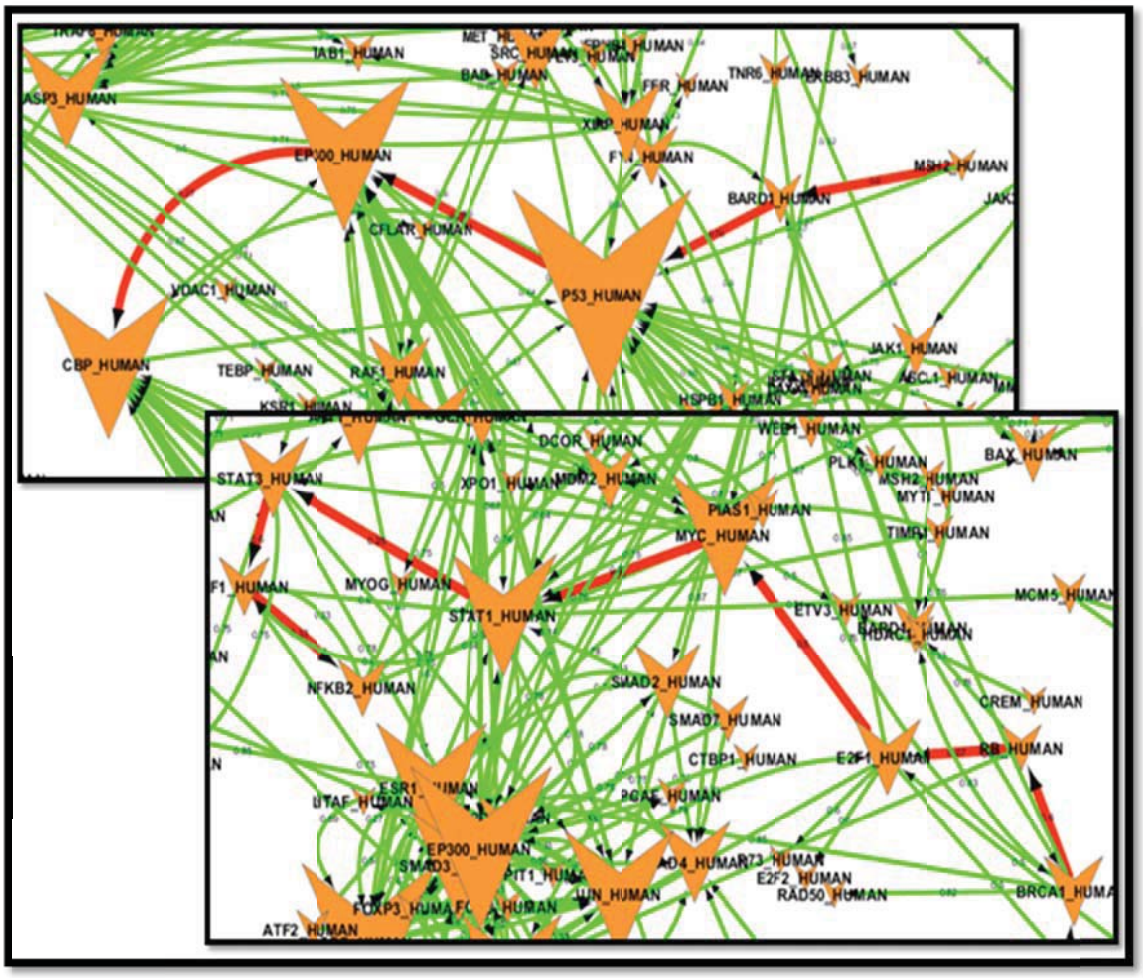


Figure 27: Maximal path derived using the proposed algorithm.

Table 7: Maximal paths derived as a result of the algorithm, Maximal path scoring and ranking at β=25%.

No.	Maximal Paths derived at σ ='2', Node weight = "degree of nodes", Edge weight = "Gene Ontology distance".	Path Score	Score1 at β=25%	Rank based on score 1
1	JAK3→JAK1→STAT1→STAT3→IRF1→NFKB2	8.5518	27	2
2	MCM5→MCM3→MCM2→MCM10	5.462	0	-
3	SRF→ELK1→MK03→MKNK1→MK14	7.98	20	3
4	FOXO1→RPP21→RBX1→SKP1→FBXW7	5.833	0	-
5	RIPK1→CRADD→CASP2→CASP9→CASP3→CASP7	9.148	38	1

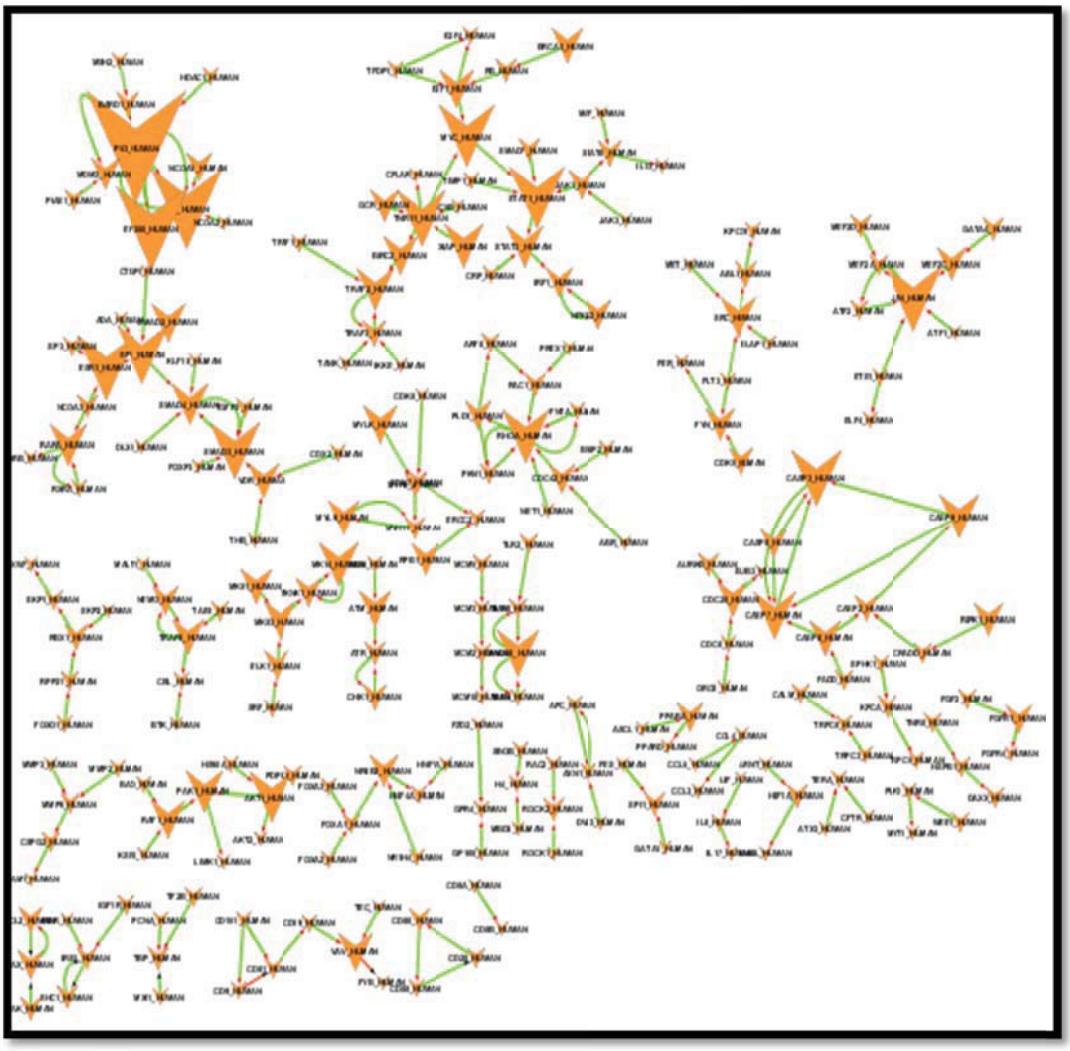


Figure 28: Subnetworks generated using the proposed algorithm.

Figure 28 shows the subnetworks derived as a result of the proposed algorithm. There were a total of 43 subnetworks. These subnetworks were validated with the functional networks generated by Metacore™. The 35 Maximal paths were also validated using Metacore™.

Validation:

The Maximal paths and the subnetworks discovered as a result of case 5(f) from section 3.4.3 were validated using Metacore™.

4.2.3.3 Network 3: (Domain 3: Biological process)

A biological process is series of events accomplished by one or more ordered assemblies of molecular functions. Examples of broad biological process terms are cellular physiological process or signal transduction. Examples of more specific terms are pyrimidine metabolic process or alpha-glucoside transport. The network in Molecular function domain has a size of 576 nodes and 1424 edges. Edges with edge weights =1 was eliminated since gene ontology distance =1 signifies "no interaction between genes". Hence the network size was reduces to 300 nodes and 449 edges.

Analysis:

We followed the procedure listed under methodology to form the canonical matrix. For this particular experiment, the threshold 'σ' was set at 2, which means the nodes with degrees less than 2 were eliminated. The subnetwork mining algorithms were carried out specifically looking for interesting maximal paths and subnetworks.

The Colorectal cancer network under the molecular function domain was mined using the proposed algorithm and 32 Maximal paths and 29 subnetworks were

discovered. The Maximal paths and subnetworks discovered were entirely different compared to that of the cellular component and molecular functions domain. Figure 29 shows the subnetworks derived as a result of the proposed algorithm. There were a total of 29 subnetworks. These subnetworks were validated with the functional networks generated by Metacore^{TM}. The 32 Maximal paths discovered were also validated using Metacore^{TM}.

Table 8 shows the Maximal paths derived as a result of the algorithm. 'Path score' is calculated using the method discussed in section 3.5(i). Score 1 is calculated using the method discussed in section 3.5(iii) at 'β=42%' (randomly assigned).

Table 8: Maximal paths derived as a result of the algorithm, Maximal path scoring and ranking at β=42%.

No.	Maximal Paths derived at σ = '2', Node weight = "degree of nodes", Edge weight = "Gene Ontology distance".	Path Score	Score 1 at β = 42%	Rank based on score1
1	PCAF→P53→SMAD3→SMAD4→CCL3	8.575	28	1
2	TSP1→FGF2→FGFR1→FGFR4→FGF6	6.24	8	3
3	NEDD8→CUL1→RBX1→SKP1→FBXW7	5.941	0	-
4	WEE1→PLK1→ANC2→CDC20	7.154	13	2
5	CXCR4→SDF1→CCL19→CCR7→GPR4→CXCR1	6.357	8	3

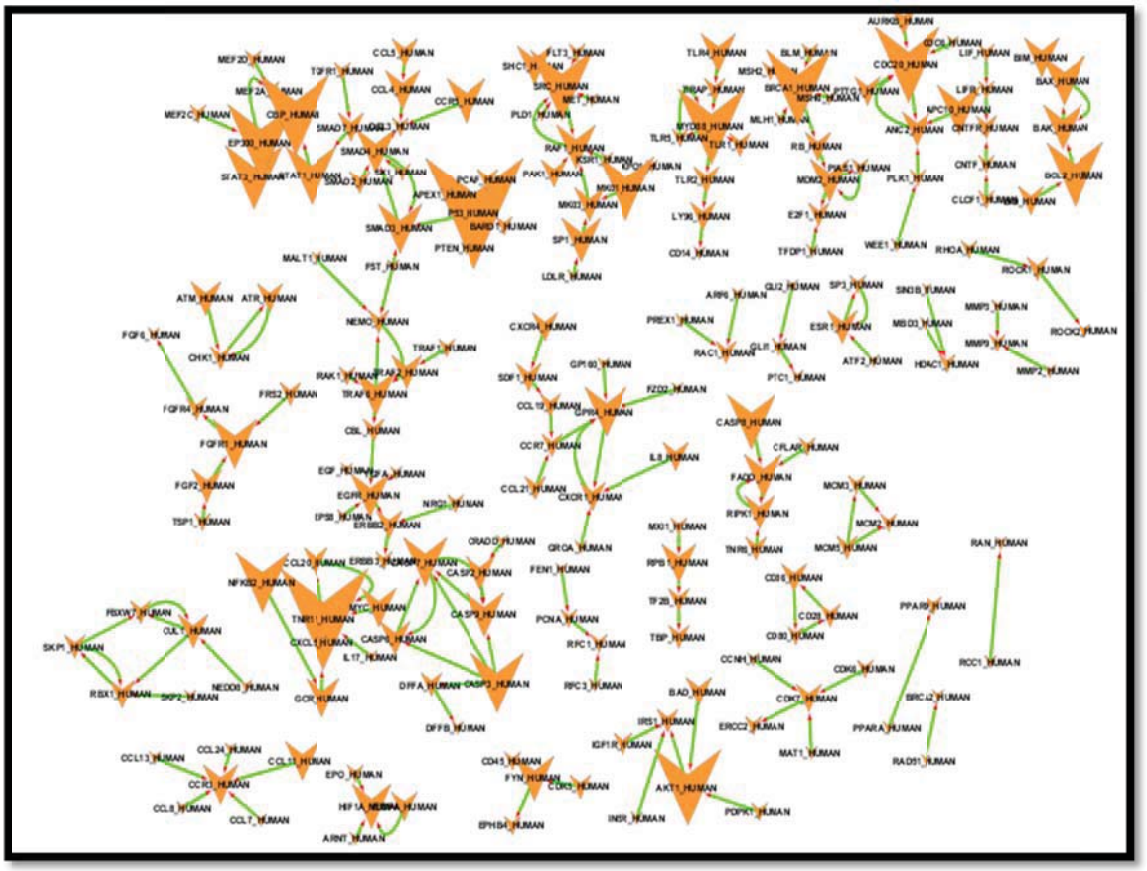


Figure 29: The subnetworks derived using the proposed algorithm.

Hence it proves that the subnetworks and the Maximal paths generated as a result of the algorithm is different in three different domains. (Cellular Components, Molecular functions and Biological Process).

4.3 Upstream and downstream of a target gene

We randomly picked a gene (P53_HUMAN) to study the significance of the gene in different domains. It was found that the interaction of each gene is different in different domains. The upstream and the downstream of the gene were different in the three cases. The Maximal paths derived using the algorithm proves that the genes interact differently in different conditions and domains.

Figure 30 shows the three Maximal paths derived in 3 different domains. Even though P53_HUMAN is present in all the three paths, the upstream and the downstream of the P53_HUMAN is different. This proves that the behavior of the genes varies in different domains resulting in dissimilar Maximal paths and subnetworks.

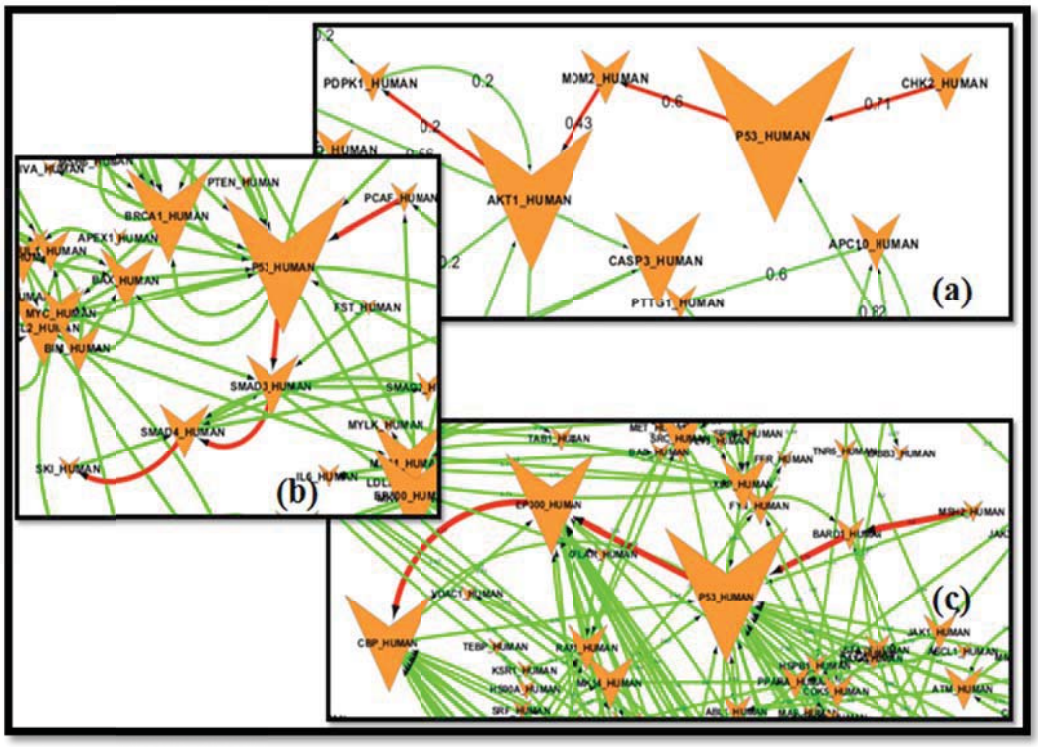


Figure 30: Maximal path comparison to differentiate between upstream and downstream of a gene in different domains. Note: (a). Maximal path derived as a result of the algorithm in Cellular Compound Domain. (b). Maximal path derived as a result of the algorithm in Biological process domain. (c). Maximal path derived as a result of the algorithm in Molecular function domain.

Validation:

The Maximal paths and the subnetworks discovered as a result of case 5(f) from section 3.4.3 were validated using Metacore^{TM}.

The results and the experimental study on both synthetic and real time datasets prove that the proposed algorithm is an efficient way to find significant Maximal paths

and subnetworks from a network given that the appropriate edge weight, node weight, and direction are provided. The results were cross verified with literature and found to be significant.

CHAPTER FIVE: DISCUSSIONS

Node weights, edge weights, and direction are the means of adding meaningful knowledge into a biological network. The node weight can define the significance of each node (genes/proteins) in the network with other nodes. The edge weight defines the pattern of interaction between the genes/proteins. The direction defines the source and the destination. This practice of adding the three parameters have suggests that it is significant in mining subnetworks and Maximal paths from the biological networks. The proposed algorithm could discover significant Maximal paths and subnetworks which were cross verified with Metacore^{TM} and literature. The Maximal paths derived using the proposed algorithm is significant and meaningful.

Preliminary study was done on a synthetic network. Significant results were obtained in the form of subnetworks and Maximal paths by testing two synthetic datasets: Facebook and rumor network. As the retrieval of subnetwork and Maximal paths solely depends on weights and direction, incorporating knowledge into node and edge is a tedious task. It was also found that the Maximal paths and the subnetworks discovered from the same set of genes but in three different domains (Chemical compounds, Biological process and Molecular functions) were different. This is a solid proof for the significance of edge weight.

Further studies were done on three sets of biological networks. Colorectal cancer dataset was the major input for the study. The gene ontology distance and degree are the two parameters which was used as edge weight and node weight in all the cases. It was found that the proposed algorithm can discover more Maximal paths from the given input data compared to Metacore™, as Metacore™ tends to miss or not identify some nodes in the network. This algorithm is also an efficient way in finding the appropriate parameters that can be used for biological Maximal path discovery. From this experiment it was postulated that gene ontology distance and degree are the significant parameters for Maximal path discovery as it ascertained a method of deducing a pathway that is reflected in the published literature. Since it is possible to change the edge weights and node weights based on a user's interest, it is a productive tool to find more Maximal paths on same network using different weights.

It was observed that most of the existing tools calculate the edge weight for a given network and then find pathways. This is a limitation compared to the proposed algorithm, as the proposed algorithm allows the user to change weights and discover different Maximal paths that are significant and meaningful.

The algorithm can also find the upstream and downstream of a given node (gene/protein/name etc.). The tool has the ability to input a single node name and the find it's upstream and downstream.

In the future work, we will be working on a better scoring mechanism compared to the one listed in this paper. We will be using Support vector machine to identify the appropriate parameters to model a network from each domain such as Gene Ontology,

Degree, Clustering Coefficient, Eccentricity, Closeness, Protein Protein Interaction (PPI) and more.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Agrawal, R. and R. Srikant, *Fast algorithms for mining association rules*, 1994, Citeseer. p. 487-499.

[2] Ahuja, R.K., et al., *Network flows: theory, algorithms and applications*, 1995, Wurzburg, Physica-Verlag, 1972-1995. p. 252-254.

[3] Albert, R. and A.L. Barabási, *Statistical mechanics of complex networks*, 2002, APS. p. 47.

[4] Albert, R., H. Jeong, and A.L. Barabási, *Internet: Diameter of the world-wide web*, 1999, Nature Publishing Group. p. 130-131.

[5] Altaf-Ul-Amin, M., et al., *Development and implementation of an algorithm for detection of protein complexes in large interaction networks*, 2006, BioMed Central Ltd. p. 207.

[6] Amaral, L.A.N., et al., *Classes of small-world networks*, 2000, National Acad Sciences. p. 11149.

[7] Ashburner, M., et al., *Gene Ontology: tool for the unification of biology*, 2000, Nature Publishing Group. p. 25-29.

[8] Babai, L. and E.M. Luks, *Canonical labeling of graphs*, 1983, ACM. p. 171-183.

[9] Bader, G.D. and C.W.V. Hogue, *An automated method for finding molecular complexes in large protein interaction networks*, 2003, BioMed Central Ltd. p. 2.

[10] Baitaluk, M., et al., *BiologicalNetworks: visualization and analysis tool for systems biology*, 2006, Oxford Univ Press. p. W466.

[11] Banks, E., et al., *NetGrep: fast network schema searches in interactomes*, 2008, BioMed Central Ltd. p. R138.

[12] Bang-J., et al., (2000), Digraphs: Theory, Algorithms and Applications, Springer, ISBN 1-85233-268-9.

[13] Barabási, A.L., R. Albert, and H. Jeong, *Scale-free characteristics of random networks: the topology of the world-wide web*, 2000, Elsevier. p. 69-77.

[14] Barabási, A.L. and Z.N. Oltvai, *Network biology: understanding the cell's functional organization*, 2004, Nature Publishing Group. p. 101-113.

[15] Biggs, N., E.K. Lloyd, and R.J. Wilson, *Graph theory, 1736-1936*1999: Oxford University Press, USA.

[16] Blatt, M., S. Wiseman, and E. Domany, *Superparamagnetic clustering of data*, 1996, APS. p. 3251-3254.

[17] Bollobás, B., *Modern graph theory*1998: Springer Verlag.

[18] Breitkreutz, B.J., C. Stark, and M. Tyers, *Osprey: a network visualization system*, 2003. p. R22.

[19] Broder, A., et al., *Graph structure in the web*, 2000, Elsevier. p. 309-320.

[20] Broido, A., *Internet topology: Connectivity of IP graphs*, 2001. p. 172.

[21] Brun, C., et al., *Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network*, 2004, BioMed Central Ltd. p. 6-6.

[22] Cakmak, A. and G. Ozsoyoglu, *Mining biological networks for unknown pathways*, 2007, Oxford Univ Press. p. 2775-2783

[23] Camacho, J., R. Guimerà, and L.A. Nunes Amaral, *Robust patterns in food web structure*, 2002, APS. p. 228102.

[24] Camp, B.H., *An Introduction to the Theory of Statistics*, 1938, JSTOR. p. 480-483.

[25] Cancho, R.F. and R.V. Solé, *The small world of human language*, 2001, The Royal Society. p. 2261.

[26] Charles, G., R. John, and H. Donal, *Predicting the response of localised oesophageal cancer to neo-adjuvant chemoradiation*.

[27] Chave, T.A., et al., *Toxic epidermal necrolysis: current evidence, practical management and future directions*, 2005, London: HK Lewis, 1963-. p. 241-253.

[28] Chen, J. and B. Yuan, *Detecting functional modules in the yeast protein–protein interaction network*, 2006, Oxford Univ Press. p. 2283.

[29] Chen, Q., et al., *The origin of power laws in Internet topologies revisited*, 2002, IEEE. p. 608-617 vol. 2.

[30] Chowell, G., J.M. Hyman, and S. Eubank, *Analysis of a real world network: The City of Portland*, 2002.

[31] Cline, M.S., et al., *Integration of biological networks and gene expression data using Cytoscape*, 2007, Nature Publishing Group. p. 2366-2382.

[32] Colak, R., et al., *Dense graphlet statistics of protein interaction and random networks*, 2009, World Scientific Pub Co Inc. p. 178.

[33] Cook, D.J. and L.B. Holder, *Substructure discovery using minimum description length and background knowledge*, 1994.

[34] Cortes, C. and V. Vapnik, *Support-vector networks*, 1995, Springer. p. 273-297.

[35] Davis, A., B.B. Gardner, and M.R. Gardner, *Deep South: A social anthropological study of caste and class*2009: Univ of South Carolina Pr.

[36] Diestel, R., *Graph theory*, 2005.

[37] Dodds, P.S. and D.H. Rothman, *Geometry of river networks. I. Scaling, fluctuations, and deviations*, 2000, APS. p. 016115.

[38] Dorogovtsev, S.N. and J.F.F. Mendes, *Evolution of networks*, 2001.

[39] Dorogovtsev, S.N. and J.F.F. Mendes, *Language as an evolving word web*, 2001, The Royal Society. p. 2603.

[40] Dost, B., et al., *QNet: A tool for querying protein interaction networks*, 2007, Springer. p. 1-15.

[41] Dunne, J.A., R.J. Williams, and N.D. Martinez, *Food-web structure and network theory: the role of connectance and size*, 2002, National Acad Sciences. p. 12917.

[42] Dunne, J.A., R.J. Williams, and N.D. Martinez, *Network structure and biodiversity loss in food webs: robustness increases with connectance*, 2002, Wiley Online Library. p. 558-567.

[43] Džeroski, S. and N. Lavra, *Relational data mining*2001: Springer Verlag.

[44] Ebel, H., L.I. Mielsch, and S. Bornholdt, *Scale-free topology of e-mail networks*, 2002.

[45] Egghe, L. and R. Rousseau, *Introduction to informetrics*1990: Elsevier Science Publishers.

[46] Enright, A.J., S. Van Dongen, and C.A. Ouzounis, *An efficient algorithm for large-scale detection of protein families*, 2002, Oxford Univ Press. p. 1575.

[47] Fahmy, S. and K. Park, *Scalability and traffic control in IP networks*, 2003, [Guildford, England; New York, NY]: IPC Science and Technology Press, c1978-. p. 203-203.

[48] Faloutsos, M., P. Faloutsos, and C. Faloutsos, *On power-law relationships of the internet topology*, 1999, ACM. p. 251-262.

[49] Fararo, T. J. and Sunshine, M.(1964). *A Study of a Biased Friendship Network*, Syracuse University Press, Syracuse.

[50] Farkas, I., et al., *The topology of the transcription regulatory network in the yeast, Saccharomyces cerevisiae*, 2003, Elsevier. p. 601-612.

[51] Fell, D.A. and A. Wagner, *The small world of metabolism*, 2000, Nature Publishing Group. p. 1121-1122.

[52] Ferro, A., et al., *NetMatch: a Cytoscape plugin for searching biological networks*, 2007, Oxford Univ Press. p. 910.

[53] Flannick, J., et al., *Graemlin: general and robust alignment of multiple large interaction networks*, 2006, Cold Spring Harbor Lab. p. 1169.

[54] Freeman, L. C. (1977). *A set of measures of centrality based on betweenness*. Sociometry **40**, 35-41

[55] Freeman, L. C. (1979). Centrality in social networks: Conceptual clarification. Social Networks, 1(3), 215-239.

[56] Fulda, S. and K.M. Debatin, *Extrinsic versus intrinsic apoptosis pathways in anticancer chemotherapy*, 2006, Nature Publishing Group. p. 4798-4811.

[57] Georgii, E., et al., *Enumeration of condition-dependent dense modules in protein interaction networks*, 2009, Oxford Univ Press. p. 933.

[58] Guelzim, N., et al., *Topological and causal structure of the yeast transcriptional regulatory network*, 2002, Nature Publishing Group. p. 60-63.

[59] Guimera, R., et al., *Self-similar community structure in organisations*, 2002.

[60] Harary, F. Graph Theory, Perseus, Cambridge, MA (1995).

[61] Hartwell, L.H., et al., *From molecular to modular cell biology*, 1999, [London: Macmillan Journals], 1869-. p. 47.

[62] Hayes, B., *Departments-Computing Science-Graph theory in practice: Part I*, 2000, New Haven, Conn.[etc.] Sigma Xi. p. 9-13.

[63] Hayes, B., *Graph theory in practice: Part II*, 2000. p. 104-109.

[64] Holme, P., C.R. Edling, and F. Liljeros, *Structure and time evolution of an Internet dating community*, 2004, Elsevier. p. 155-174.

[65] Hu, H., et al., *Mining coherent dense subgraphs across massive biological networks for functional discovery*, 2005, Oxford Univ Press. p. i213.

[66] Hu, Z., et al., *VisANT 3.0: new modules for pathway visualization, editing, prediction and construction*, 2007, Oxford Univ Press. p. W625.

[67] Huan, J., W. Wang, and J. Prins, *Efficient mining of frequent subgraphs in the presence of isomorphism*, 2003, Published by the IEEE Computer Society.

[68] Inokuchi, A., T. Washio, and H. Motoda, *Complete mining of frequent patterns from graphs: Mining graph data*, 2003, Springer. p. 321-354.

[69] Ito, T., et al., *A comprehensive two-hybrid analysis to explore the yeast protein interactome*, 2001, National Acad Sciences. p. 4569.

[70] Jeong, H., et al., *Lethality and centrality in protein networks*, 2001, Nature Publishing Group. p. 41-42.

[71] Jeong, H., et al., *The large-scale organization of metabolic networks*, 2000, Nature Publishing Group. p. 651-654.

[72] Jordano, P., J. Bascompte, and J.M. Olesen, *Invariant properties in coevolutionary networks of plant–animal interactions*, 2003, Wiley Online Library. p. 69-81.

[73] Kalaev, M., et al., *NetworkBLAST: comparative analysis of protein networks*, 2008, Oxford Univ Press. p. 594.

[74] Kalapala, V.K., V. Sanwalani, and C. Moore, *The structure of the United States road network*, 2003.

[75] Kauffman, S.A., *Metabolic stability and epigenesis in randomly constructed genetic nets*, 1969, Elsevier. p. 437-467.

[76] Kauffman, S.A., *Gene regulation networks: A theory for their global structure and behaviors*, 1977, Academic Press. p. 145–182.

[77] Kauffman, S.A., *The origins of order*. Vol. 209. 1993: Oxford University Press New York, NY.

[78] Kelley, B.P., et al., *Conserved pathways within bacteria and yeast as revealed by global protein network alignment*, 2003, National Acad Sciences. p. 11394.

[79] Kermarrec, A.M., et al., *Second order centrality: Distributed assessment of nodes criticity in complex networks*, Elsevier.

[80] King,A.D. *et al.* (2004) An efficient algorithm for large-scale detection of protein families. *Bioinformatics*, **20**, 3013–3020.

[81] Kinouchi, O., et al., *Deterministic walks in random networks: An application to thesaurus graphs*, 2002, Elsevier. p. 665-676.

[82] Kleinberg, J.M., et al., *The web as a graph: Measurements, models, and methods*, 1999, Springer-Verlag. p. 1-17.

[83] Knuth, D.E., *The Stanford GraphBase: a platform for combinatorial computing*1993: AcM Press.

[84] Koyutürk, M., et al., *Detecting conserved interaction patterns in biological networks*, 2006, Mary Ann Liebert, Inc. 2 Madison Avenue Larchmont, NY 10538 USA. p. 1299-1322.

[85] Kumar, K., et al., *BioMap: toward the development of a knowledge base of biomedical literature*, 2004, ACM. p. 121-127.

[86] Kuramochi, M. and G. Karypis, *Frequent subgraph discovery*, 2001, Published by the IEEE Computer Society. p. 313.

[87] Lacroix, V., C.G. Fernandes, and M.F. Sagot, *Motif search in graphs: application to metabolic networks*, 2006, Published by the IEEE CS, CI, and EMB Societies & the ACM. p. 360-368.

[88] Latora, V. and M. Marchiori, *Is the Boston subway a small-world network?*, 2002, Elsevier. p. 109-113.

[89] Lee, T.I., et al., *Transcriptional regulatory networks in Saccharomyces cerevisiae*, 2002, American Association for the Advancement of Science. p. 799.

[90] Lin, C.Y., et al., *Hubba: hub objects analyzer—a framework of interactome hubs identification for network biology*, 2008, Oxford Univ Press. p. W438.

[91] Loewenstein, Y., et al., *Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space*, 2008, Oxford Univ Press. p. i41.

[92] Lorrain, F. and H.C. White, *Structural equivalence of individuals in social networks*, 1971, Routledge. p. 49-80.

[93] Mariolis, P., *Interlocking directorates and control of corporations: The theory of bank control*, 1975. p. 425-439.

[94] Maritan, A., et al., *Scaling laws for river networks*, 1996, APS. p. 1510.

[95] Maslov, S. and K. Sneppen, *Specificity and stability in topology of protein networks*, 2002, American Association for the Advancement of Science. p. 910.

[96] McGovern, A. and D. Jensen, *Chi-squared: A simpler evaluation function for multiple-instance learning*, 2003, Citeseer.

[97] Milgram, S., *The small world problem*, 1967, New York. p. 60-67.

[98] Montoya, J.M. and R.V. Solé, *Small world patterns in food webs*, 2002, Elsevier. p. 405-412.

[99] Moody, J. and D.R. White, *Structural cohesion and embeddedness: A hierarchical concept of social groups*, 2003, JSTOR. p. 103-127.

[100] Moreno, J.L., *Who shall survive?*1953: JSTOR.

[101] Motter, A.E., et al., *Topology of the conceptual network of language*, 2002, APS. p. 065102.

[102] Myers, C., et al., *Discovery of biological networks from diverse functional genomic data*, 2005, BioMed Central Ltd. p. R114.

[103] Navlakha, S., M.C. Schatz, and C. Kingsford, *Revealing biological modules via graph summarization*, 2009, Mary Ann Liebert, Inc. 2 Madison Avenue Larchmont, NY 10538 USA. p. 253-264.

[104] Newman, M.E.J., *Models of the small world*, 2000, Springer. p. 819-841.

[105] Newman, M.E.J., *The structure and function of complex networks*, 2003, JSTOR. p. 167-256.

[106] Newman, M.E.J., C. Moore, and D.J. Watts, *Mean-field solution of the small-world network model*, 2000, APS. p. 3201-3204.

[107] Newman, M.E.J. and D.J. Watts, *The structure and dynamics of networks*2006: Princeton Univ Pr.

[108] Nikolsky, Y., et al., *A novel method for generation of signature networks as biomarkers from complex high throughput data*, 2005, Elsevier. p. 20-29.

[109] Opsahl, T., F. Agneessens, and J. Skvoretz, *Node centrality in weighted networks: Generalizing degree and shortest paths*, Elsevier. p. 245-251.

[110] Ovaska, K., M. Laakso, and S. Hautaniemi, *Fast Gene Ontology based clustering for microarray experiments*, 2008, Springer. p. 1-8.

[111] Padgett, J.F. and C.K. Ansell, *Robust Action and the Rise of the Medici*, 1993. p. 1259-1319.

[112] Palla, G., et al., *Uncovering the overlapping community structure of complex networks in nature and society*, 2005, Nature Publishing Group. p. 814-818.

[113] Pandey, J., et al., *Functional annotation of regulatory pathways*, 2007, Oxford Univ Press. p. i377.

[114] Pandian, T.J., *Annual Review of Earth and Planetary*, 2003. p. 824.

[115] Parikka, P., et al., *Pathway Assistant: a web portal for metabolic modelling*, 2008, Citeseer.

[116] Pattabhiraman, S., *Transcriptional regulation of 12/15-lipoxygenase expression and the implication of the enzyme in hepoxilin biosynthesis and apoptosis*, 2003, Humboldt-Univ.

[117] Pereira-Leal, J.B., A.J. Enright, and C.A. Ouzounis, *Detection of functional modules from protein interaction networks*, 2001, Citeseer. p. 242-245.

[118] Pinter, R.Y., et al., *Alignment of metabolic pathways*, 2005, Oxford Univ Press. p. 3401.

[119] Podani, J., et al., *Comparable system-level organization of Archaea and Eukaryotes*, 2001, Nature Publishing Group. p. 54-56.

[120] Rapoport, A., *Contribution to the theory of random and biased nets*, 1957, Springer. p. 257-277.

[121] Rapoport, A. and W.J. Horvath, *A study of a large sociogram*, 1961, Wiley Online Library. p. 279-291.

[122] Reimand, J., et al., *GraphWeb: mining heterogeneous biological networks for gene modules with functional significance*, 2008, Oxford Univ Press. p. W452.

[123] Rives, A.W. and T. Galitski, *Modular organization of cellular networks*, 2003, National Acad Sciences. p. 1128.

[124] Rodriguez-Iturbe, I. and A. Rinaldo (1998). Channel networks, Annual Review of Earth and Planetary Science 26, 289–327.

[125] Rodriguez-Iturbe, I. and A. Rinaldo, *Fractal river basins: chance and self-organization*2001: Cambridge Univ Pr.

[126] Roethlisberger, F.J. and W.J. Dickson, *Management and the Worker*, 1939.

[127] Samanta, M.P. and S. Liang, *Predicting protein functions from redundancies in large-scale protein interaction networks*, 2003, National Acad Sciences. p. 12579.

[128] Schreiber, F. and H. Schwöbbermeyer, *MAVisto: a tool for the exploration of network motifs*, 2005, Oxford Univ Press. p. 3572.

[129] Scott, J., *Social network analysis*, 2000.

[130] Sen, P., et al., *Small-world properties of the Indian railway network*, 2002.

[131] Sharan, R., et al., *Conserved patterns of protein interaction in multiple species*, 2005, National Acad Sciences. p. 1974.

[132] Shen-Orr, S.S., et al., *Network motifs in the transcriptional regulation network of Escherichia coli*, 2002, Nature Publishing Group. p. 64-68.

[133] Shlomi, T., et al., *QPath: a method for querying pathways in a protein-protein interaction network*, 2006, BioMed Central Ltd. p. 199.

[134] Sigman, M. and G.A. Cecchi, *Global organization of the Wordnet lexicon*, 2002, National Acad Sciences. p. 1742.

[135] Skiena, S., *Implementing discrete mathematics: combinatorics and graph theory with Mathematica*1991: Addison-Wesley Longman Publishing Co., Inc.

[136] Smith, R.D., *Instant messaging as a scale-free network*, 2002.

[137] Sohler, F. and R. Zimmer, *Identifying active transcription factors and kinases from expression data using pathway queries*, 2005, Oxford Univ Press. p. ii115.

[138] Sole, R.V. and M. Montoya, *Complexity and fragility in ecological networks*, 2001, The Royal Society. p. 2039.

[139] Solé, R.V. and R. Pastor Satorras, *Complex networks in genomics and proteomics*, 2002, Wiley Online Library.

[140] Spirin, V. and L.A. Mirny, *Protein complexes and functional modules in molecular networks*, 2003, National Acad Sciences. p. 12123.

[141] Sporns, O., *Network analysis, complexity, and brain function*, 2002, Wiley Online Library. p. 56-60.

[142] Sporns, O., G. Tononi, and G.M. Edelman, *Theoretical neuroanatomy: relating anatomical and functional connectivity in graphs and cortical connection matrices*, 2000, Oxford Univ Press. p. 127.

[143] Stelling, J., et al., *Metabolic network structure determines key aspects of functionality and regulation*, 2002, Nature Publishing Group. p. 190-193.

[144] Steyvers, M. and J.B. Tenenbaum, *The Large Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth*, 2005, Wiley Online Library. p. 41-78.

[145] Strogatz, S.H., *Exploring complex networks*, 2001, Nature Publishing Group. p. 268-276.

[146] Tague-Sutcliffe, J., *An introduction to informetrics*, 1992, Elsevier. p. 1-3.

[147] Travers, J. and S. Milgram, *An experimental study of the small world problem*, 1969, JSTOR. p. 425-443.

[148] Opsahl, T., et al. *Node centrality in weighted networks Generalization degree and shortest paths*. Social Networks 2010 32: 245.

[149] Uetz, P., et al., *A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae*, 2000, Nature Publishing Group. p. 623-627.

[150] Ulitsky, I. and R. Shamir, *Identification of functional modules using network topology and high-throughput data*, 2007, BioMed Central Ltd. p. 8.

[151] Vertigan, D. and G. Whittle, *A 2-isomorphism theorem for hypergraphs*, 1997, Academic Press, Inc. p. 215-230.

[152] Wagner, A. and D.A. Fell, *The small world inside large metabolic networks*, 2001, The Royal Society. p. 1803.

[153] Wascholowski, V. and A. Giannis, *Neutral sphingomyelinase as a target for drug design*, 2001. p. 581-90.

[154] Wasserman, S. and K. Faust, *Social network analysis: Methods and applications*1995: Cambridge university press.

[155] Watts, D.J., *Small worlds: the dynamics of networks between order and randomness*2003: Princeton Univ Pr.

[156] Watts, D.J. and S.H. Strogatz, *Collective dynamics of 'small-world'networks*, 1998, Nature Publishing Group. p. 440-442.

[157] Wernicke, S. and F. Rasche, *FANMOD: a tool for fast network motif detection*, 2006, Oxford Univ Press. p. 1152.

[158] White, J.G., et al., *The structure of the nervous system of the nematode Caenorhabditis elegans*, 1986, The Royal Society. p. 1.

[159] Williams, R.J., et al., *Two degrees of separation in complex food webs*, 2002, National Acad Sciences. p. 12913.

[160] Yan, X. and J. Han, *gSpan: Graph-based substructure pattern mining*, 2002, Published by the IEEE Computer Society.

[161] Yip, K.Y., et al., *The tYNA platform for comparative interactomics: a web tool for managing, comparing and mining multiple networks*, 2006, Oxford Univ Press. p. 2968.

[162] Yoshida, K., H. Motoda, and N. Indurkhya, *Graph-based induction as a unified learning framework*, 1994, Springer. p. 297-316.

[163] Availablefrom: http://www.sabiosciences.com/pathway.php?sn=Cellular_Apoptosis_Pathway.