The Hong Kong Polytechnic University
Department of Computing

MINING STRUCTURAL PATTERNS IN BIOLOGICAL NETWORKS

Lam Wai Man

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

August, 2009

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

LAM WAI MAN (Name of student)

Abstract

Biological networks capture information about the way biomolecules, such as genes, proteins, and metabolites, interact with each other. Discovering interesting patterns in them will enable one to better understand biological processes such as cellular organization, transcription regulation and phenotypic evolution, etc. Biological networks can be modeled as graphs with vertices representing biomolecules and edges representing the interactions between them. Graph mining algorithms have been used to find frequently-occurring subgraphs in biological graphs. As these subgraphs may only be "overrepresented patterns", these algorithms are sometimes not considered very useful. What is needed is an algorithm that can be used to find not only frequently-occurring patterns, but patterns that can actually characterize biological networks and allow them to be discriminated from each other.

For many biological networks, other than the name of each of their constituent biomolecules, a number of other attributes are usually also known about them. For example, other than the name of each protein in a PPI network, we also know, for many of the proteins, the functions they perform, the cellular processes they are involved in, etc. Proteins always perform more than one molecular function and are involved in multiple cellular processes [67]. The information provided by all these additional attributes are currently not taken into consideration by graph mining algorithms even though they can be very useful. To take into considerations the multiple attributes of the constituent biomolecules, we model the biological network as a multiple-attribute graph using gene ontology to allow more information, other

than direct interactions between biomolecules, to be used in the graph mining process. The multiple-attribute graph representation allows vertices to not only represent biomolecules but also the attributes that associate with them. Subgraphs in a multiple-attribute graph may relate to each other and if a node is used to represent a subgraph, hierarchical multiple attribute graph can also be formed and mined for patterns. In this thesis, we propose a graph mining algorithm that can be used to discover interesting patterns in such graphs. The algorithm is called *MISPAG* (Mining Interesting Structural Patterns in Attributed Graphs). *MISPAG* is able to discover interesting subgraphs using an interestingness measure that can be used to determine if a certain subgraph occurs more, or less, frequently in a graph than expected. The interestingness measure can take into consideration the multiple attributes of the constituent biomolecules of a biological network and can be used to filter out subgraphs that do not contribute to the unique characterization and discrimination of a network or a class of networks even if they occur frequently according to some user threshold. *MISPAG* can be modified as different algorithms that suitable to solve such problems as motif discovery, network identification, protein function prediction, molecular classification, and protein complexes discovery. These algorithms have been implemented and tested with real biological data in different application areas. Experimental results show that our proposed algorithms can effectively uncover patterns that are biologically meaningful for the deciphering of the biological and structural relationships in the networks, and for the prediction of un-annotated functions and features of proteins, genes, and chemical compounds.

Publications arising from the Thesis

- Winnie W. M. Lam, and Keith C. C. Chan, A Graph Mining Algorithm for *Class*ifying Chemical Compounds, *IEEE International Conference on BioInformation and BioMedicine* (BIBM '08), pp.321-324, 2008. (Accepted)
- Winnie W. M. Lam, and Keith C. C. Chan, Discovering Interesting Molecular Substructures from Molecular *Class*ification, *IEEE Transaction on NanoBioscience*. (Accepted)
- Winnie W.M. Lam, and Keith C.C. Chan, Discovering Interesting Patterns from the PPI Network of Saccharomyces cerevisiae, *BMC Bioinformatics*. (Submitted)
- Winnie W. M. Lam, and Keith C. C. Chan, A Hybrid Neighbor Approach for Function Prediction in Biological Networks (Submitted).
- Winnie W. M. Lam, and Keith C. C. Chan, Enhanced Feature-Based Classification of Protein-Protein Interaction Networks, *IEEE Transactions on Biomedical Engineering* (TBE). (Submitted).

Acknowledgements

First and foremost I would like to thank my supervisor, Prof. Keith C. C. Chan, for his guidance, patience, and invaluable advice. His continuous support and valuable feedback contributed greatly to my research study. I deeply express my sincere appreciation for his great supervision.

Further, I must extend my sincere thanks to my parents and friends. They gave me their full support and encouragement when I was frustrated and loss of confidence. Without their love and supports, I would not be able to complete my study.

Table of contents

Chapter 1

Introduction

Due to advances in technologies in obtaining complete genome sequences and high-throughput post-genomic experimental data, biological network are made available to the general public [1] through such databases as MIPS [14], DIP [15], BioGRID [16], STRING [17], MINT [68], IntAct [49], HPRD [76], KEGG [70], MetaCyc [77], Reactome [78], GIN [79], GeneNet [80], ITFP [81], etc.

Biological networks such as protein interaction networks, gene regulatory networks, metabolic networks, and phylogenetic networks, etc. are highly complex, and the discovery of structural patterns in such networks are essential to the understanding network topologies and how they can influence the functioning and evolution of biological systems. Biomolecules such as genes, proteins, and metabolites are expected to interact with each other in these biological networks. The discovered patterns may reveal interaction patterns among these biomolecules to allow biologically interesting concepts such as common network motifs, evolutionary relationships among species, organization of functional modules, etc., to be identified. The discovered patterns may also lead to better understanding of such biological processes as cellular organization, transcription regulation and phenotypic evolution.

Given that biological network data are now more widely available, a number of studies [4, 6, 71, 96] have been conducted to mine different types of

biological networks, such as protein interaction networks, gene regulatory networks, metabolic networks, and phylogenetic networks, etc. for interesting structural patterns. There have also been attempts to correlate these patterns with such topological entities [99] as degree distribution, average clustering co-efficient, average path length and centrality, functional roles, possible distortions of interactions of the network and modification of biomolecules so as to understand why and how a particular pattern may lead to the development of certain diseases and provide the basis for new therapeutic approaches [2, 3, 5]. For example, genes that are related to similar disease phenotypes are found to be likely to be functionally related, such as participating in a common pathway or signal transduction mechanism [104, 105]. The cancer proteins are found to be highly connected with other cancer-related proteins [106], and a study of the PPI network of herpesvirus [107] indicates that viral networks differ significantly from cellular networks, which raises the hypothesis that other intracellular pathogens might also have distinguishing topologies.

Biological networks can be modeled as graphs with vertices and edges representing, respectively, the biomolecules and interactions among them. Given a number of biological networks represented in graphs, one most common approach to discover structural patterns in these graphs is to use graph mining algorithms [40, 45, 74, 75]. These algorithms have been used to discover frequently-occurring subgraphs using a candidate generation process to enumerate frequent subgraphs. To do so, they rely typically on the use of simple a priori and conditional probabilities names support and confidence measure to decide if a subgraph occurrs frequently. If it is,

then the subgraph can represent a network motif. Graph mining algorithms are sometimes not considered the most useful [101, 102] as the frequent subgraphs discovered may only represent "overrepresented patterns" [103]. If a user chooses a relatively small support threshold, a large number is usually found by these graph mining algorithms. However, if the support threshold is made larger, it is also possible for too little patterns to be found. The support threshold that controls the frequency that a subgraph has to appear in a larger graph is usually hard to determine and they have to be discovered by trials and errors.

When discovering useful structural patterns in biological networks, a subgraph that occur frequently enough does not always mean that it is interesting. In fact a network motif, for example, is often not just a frequently-occurring subgraph. It is a subgraph that occurs more frequently than expected, i.e., it is the relative, rather than the absolute frequency of occurrence that matters. For example, the feed-forward loops (i.e. X→Y→Z→X) occurs much more frequently in the transcriptional regulation network of *Escherichia coli* than in randomized networks, and sets of genes in such network that are regulated by different transcription factors are overlaps much more than expected if it only occurs at random [108].

Since the discovery of such structural patterns as network motifs can reveal how they interact with each other functionally, it is important to develop an effective algorithm to mine for such patterns. Some interesting structural patterns, in the form of motifs, exhibiting certain dynamical behavior, have been identified as essential ingredients of specific biological processes [100]. Such algorithm has to identify

patterns that can be used both for characterization of biological networks and also for discrimination among them so that they can represent each network uniquely and can allow the structural characteristics of different networks to be differentiable easily from the others. These patterns may not appear frequently enough to be discovered by existing graph mining algorithm but they are interesting and important. In addition to being able to identify such interesting patterns, the algorithm that we need has to be able to handle multiple attributes of the biomolecules in a network. This is because, as we discussed above, for many biological networks, other than the name of each of their constituent biomolecules, a number of other attributes such as molecular function, cellular component, biological process, etc. are usually also known about them. Also, many constituent biomolecules are involved in multiple cellular processes and performed more than one molecular function. To improve the effectiveness of graph mining, these functional attributes can also be included in the graph annotation. As the existing graph mining algorithms are mainly developed to tackle graphs involving single-attribute vertices and edges, they do not take additional attributes into considerations even though they can be very useful. In order to discover interesting patterns in biological networks, there is a need that biological networks be represented as graphs that can allow vertices and edges have multiple attributes so that more information about the constituent biomoleules can be considered during the mining process.

1.1 Problem Statements

Given one or more biological networks that is each made up of a number of interacting biomolecules, and given that each such biomolecule is characterized by a set of attributes such as its name, physical properties, functions, etc, there have been some attempts to represent these biological networks as graphs so that the vertices represent biomolecules and the edges represent the interactions among them. For example, in a protein-protein interaction (PPI) network, the proteins are represented as vertices and the existence of an interaction relationship between two proteins is represented as an edge between their corresponding vertices.

Since a set of attributes is associated with each biomolecule, there is a need for these attributes to be captured and represented in a biological network graph as well. In the case of PPI networks, for example, a protein is associated with a set of attributes for that protein. These attributes can represent the physical and chemical properties of the protein, the domain that the protein belongs to, the molecular function that it performs, the biological process that the protein is involved in, and the cellular component that it is located in, etc. In a similar way, different types of interactions, such as physical, chemical, biochemical, and a hybrid of interactions, can exist between two protein molecules. The inclusion of these attributes in the analysis of biological networks can provide important information about the underlying patterns. One problem that we intend to tackle is how these different attributes can be captured in the graph that represent these biological networks and how such a graph can be analyzed so that hidden patterns can be discovered in them.

Given one or more biological networks with the attributes that associate with each biomolecule in these networks captured in one or more corresponding multiple-attribute graphs, one main problem that we are concerned with is to discover interesting structural patterns in these graphs. Unlike many graph mining algorithms, the interestingness of a pattern is not defined here to be subgraphs that appear more frequently than a user-specified threshold. Instead, it is to be defined as subgraphs that can allow one graph or one set of graphs to be characterized for easy discrimination against the others. Since a subgraph which is frequent in one graph may also be frequent in another, there is a need for a graph mining algorithm to discover patterns that are discriminative. If such patterns can be discovered, they can be used to tackle the problems of the discovery of structural motifs in PPI networks, the prediction of un-annotated protein functions, the classification of biological networks, the discovery of class-specific patterns from molecular data, and the identification of protein complexes in PPI networks, etc. If an effective approach can be developed to discover hidden structural patterns in biological networks, these problems that can be addressed much more easily.

Discovery of motifs is concerned with the discovery of frequently occurring structures or of structures that occur more frequently than random. Given a PPI network, for example, its proteins and interactions can be represented as vertices and edges in a graph respectively. Since PPI networks can be very complex, their graphs can contain tens of thousands of vertices and edges, a graph mining algorithms is needed to allow us to identify subgraphs called motifs and in this thesis, such an algorithm will be developed. The discovery of sturcutral motifs can be used for the

description of the structure and functionality of a PPI network. It may uniquely characterize a network and even a class of networks [97]. For example, by analyzing the relatedness of protein functions in PPI networks across different species, one may be able to discover a set of interesting subgraphs that can characterize a class of species, and this can lead to better understanding of such biological processes as cellular organization and transcription. Certain motifs that exhibit dynamical functions have in fact be identified as essential component in specific biological processes [98].

Given a partially annotated PPI network, it is also the intention of this thesis to present an approach to predict the functions of the un-annotated proteins in it. Since the number of proteins in a PPI network is so large, it is very expensive to determine the functions of all proteins experimentally. Within a PPI network, proteins are not working alone; instead they interact with the other neighbors to perform certain functions. As the function of a protein can be inferred by analyzing its neighbors, we can make use of this characteristic to develop computational approaches to predict the unknown functions. The prediction of un-annotated protein functions from PPI networks is possible with an effective approach to discovering structural patterns in multiple-attribute network graphs. As proteins are responsible for many different important functions in living cells, an effective solution to the problem can allow better understanding the molecular and biochemical processes that sustain health and cause disease, and the biologists can more easily design molecules that bind to the proteins and obtain better drugs

Given a set of biological networks with each pre-classified to two or more classes, the problem of discovering sturtual patterns in each of these classes for biological network classification is a challenging task. If patterns that can allow each class of networks to be uniquely chracterizied and discriminated against each other, it can be used to predict the class membership of a biological network that is not originally given in the data set. Given an effective way of discovering interesting subgraphs in multiple-attribute network graph, this problem can be tackled effectively to allow biological network to be grouped with those that it is related to.

One application of an effective network classification algorithm is to use it to classify molecular data. Many molecular databases have been made available online, and there is an increasing need for techniques to be developed to mine these data for interesting patterns for molecular classification. Given a set of molecular structure data pre-classified into a number of classes, we can discover class-specific patterns so that “unseen” molecules not originally in the data set can be accurately classified. The discovery of class-specific patterns from molecular data can decipher the functional characteristics of different classes of molecules. The molecular databases store the information that defines the physical and chemical properties which is essential for drug discovery. To understand how the atoms in a molecule are interrelated and how different molecules are differentiated, the identification of common structure between molecules is not enough. By modeling each molecule as a molecular graph, we can discover the class-specific patterns for each class of molecules by discovering interesting patterns with discriminative power. These

interesting patterns can uniquely define each class with certain degree of interestingness and further be used for classifying unknown molecules.

Proteins usually perform functions in a group of two or more proteins and they are called protein complexes [154, 155], and such protein complexes play important roles in cells, Many computational algorithms have been developed to explore protein complexes from PPI networks. Given a PPI network represents as a graph structure, the algorithms of identifying protein complexes is modeling as the problem of graph clustering algorithm. The given network graph that contains a set of protein complexes is clustered to form groups of protein complexes, The identification of protein complexes in PPI networks helps us to understand how proteins are interacting with each other to perform functions in a group. Such protein complexes are important to be identified as they play many important roles in cells. However, the number of experimentally-determined protein complexes is still far from complete, so there is a crucial need to develop an effective computational method to accurately identify such protein complexes from large-scale PPI networks.

1.2 Overview of Solutions

In the last section, we described the problems that we intend to tackle in this thesis, i.e., the discovering of structural patterns in one or more multiple-attribute graphs for the characterization and discrimination of one or more biological networks. To tackle the problem, we need a graph representation that can take into consideration the various attributes of the biomolecules that make up the biological networks. We also

need an interestingness measure to allow us to decide if a structural pattern is useful for such purpose. In addition, we need an algorithm that makes use of the interestingness measure to actually discover interesting patterns and to use such patterns to tackle the problems of motif discovery, protein function prediction, classification of biological networks and molecular data, and protein-complex identification, etc.

First of all, each biological network will be modeled as a network graph that captures the structural information of different relations between biomolecules. For each such biomolecule, a number of attributes are usually known. To take into consideration these multiple attributes when mining the networks, multiple-attribute (MA) graph is introduced. To represent a network as a MA graph, the vertices and edges are allowed to have multiple attributes and each vertex and edge can have multiple values corresponding to each attribute. Such a scheme can be used to represent any complex network structure.

Given such a set of MA graphs corresponding to a set of biological networks, it is important to discover biologically meaningful patterns. To discover such patterns, we propose an algorithm called *MISPAG* to discover interesting subgraphs with different degrees of interestingness by an interestingness measure. *MISPAG* defines an interestingness measure to find interesting structural patterns in one or more biological graphs by determining if a certain pattern occurs more, or less, frequently in a graph than expected. It can objectively determine the interestingness score of subgraphs by comparing their conditional and apriori probabilities.

Given a set of biological networks pre-classified into a number of classes, a number of graph mining algorithms have been used to discover useful subgraphs in such network data. However, they usually discover subgraphs independently for each class, and the number of discovered subgraphs is generally large in many real biological networks. By using the interestingness measure used in *MISPAG*, *MISPAG-FP* is proposed to filter out uninteresting patterns, which can significantly reduce the number of subgraphs that needs to be considered for various graph mining applications.

The interestingness measure used in *MISPAG* can be used in another algorithm called *MISPAG-DM* to discover motifs from a set of functionally annotated network graphs such as those from PPI networks. To do so, each protein is represented as a vertex and each interaction between two proteins is represented as an edge for functional annotation. To discover interesting structural patterns, we determine the interestingness of the interactions. Each edge in the network graph will be assigned with a degree of interestingness, and only those interesting edges are expanded to determine if it is part of an interesting subgraph. With such an algorithm, we are able to discover subgraph patterns that are biologically meaningful. The discovered interesting subgraphs provide positive or negative evidence supporting or refuting the classification of a graph into a particular class of network.

To discover missing functions of biomolecules in many real biological networks, we make use of the interestingness measure used in MISPAG to predict biomolecular functions in a network using an algorithm called *MISPAG-PF*. The

network graph is first transformed into a hybrid network graph that contains the information of both direct and shared neighbors. The function associations between a biomolecule and its direct and shared neighbors are defined by the interestingness score. *MISPAG-PF* can identify all interesting associations between biomolecules for function prediction, and helps to solve the problem that some biomolecules may direct or indirect interact with related genes instead of common genes.

Based on *MISPAG*, we have also developed *MISPAG-MA* which can be used to address the problem of mining patterns from biological networks among different species. It can discover interesting patterns that can represent different levels of network organization and provide true characterization among different networks. The networks are represented as multiple-attribute graphs by annotating each biomolecule with multiple attributes. *MISPAG-MA* allows multiple attributes to be associated with vertices and edges in a graph. It is able to discover interesting patterns in such graphs to allow each of them to be both characterized and discriminated from the others. We measure how interesting these subgraphs are with a weight-of-evidence that is provided by each individual interesting frequent subgraph for a given graph to be classified into a class.

The structure of molecules stores the information that defines the functions of their physical and chemical properties. In another algorithm based on MISPAC, we represent each compound as a hierarchical attributed graph with multiple levels of complexity. *MISPAG-CM* is proposed to identify interesting components from each

class of compounds in the molecular database, and these interesting components are class-specific and useful for classifying of molecular structures.

To discover interesting patterns from biological networks that correspond to biologically meaningful patterns such as protein complexes, *MISPAG-PC* is proposed. As protein usually perform functions in groups of two or more, we can use *MISPAG-PC* to discover functionally related proteins that can potentially form protein complexes. The key idea of *MISPAG-PC* is to capture biological relationships between proteins, and define a significant score for each interaction for identifying interesting subgraphs as protein complexes. Proteins in a complex are not only structurally inter-connected, but are also biologically related. The PPI network graph is labeled with known molecular functions, and each interaction between two neighboring proteins is assigned with an interestingness score by *MISPAG-PC*. The maximum significant score is first obtained for each protein, and the interactions with significant score lower than the threshold will be filtered. After filtering those uninteresting interactions, the connected interesting subgraphs will be extracted from the remaining graph as predicted protein complexes.

In summary, the different versions of *MISPAG* are useful for various graph mining tasks for biological networks for filtering uninteresting patterns (*MISPAG-FP*), discovering motifs (*MISPAG-DM*), predicting unknown protein functions (*MISPAG-PF*), mining interesting patterns from multiple-attribute networks (*MISPAG-MA*), classifying of molecular structures (*MISPAG-CM*), and clustering networks for biologically meaningful protein complexes (*MISPAG-PC*). These

algorithms have been tested with real data and the results indicate that they can be very useful in dealing with the problems that they are developed for.

1.3 Thesis Organization

In the next chapter, the basic concepts of biological interaction network and and the existing computational approaches for analyzing the structural information of the biological interaction networks is described. We also discussed how they can be improved.

Chapter 3 presents the details of the adjusted residual as an interestingness measure for discriminating interesting patterns from the uninteresting ones in the biological networks. Many existing graph mining algorithms generate a large number of subgraphs that frequently occur in a graph but not all of them are useful. We applied our approach to such graph mining algorithms to illustrate its effectiveness to identify the interesting ones from the discovered subgraphs.

Chapter 4 describes the discovery of statistically significant motifs from PPI networks based on the interestingness measure. Both structural and functional information of the network are considered to discover such interesting structural patterns. We describe in this chapter how we first annotate each protein in the network with its corresponding molecular function, and then the interesting interactions are extracted to form interesting subgraphs. The experimental results show that we can discover structural patterns, such as protein complexes, that may

not occur frequently enough to be discovered by graph mining algorithms but are interesting and biologically meaningful.

Chapter 5 presents how a hybrid network graph model we propose can be useful in the representation of different associations in gene interaction networks. By considering the hybrid neighborhood concept with the interestingness measure, we can predict functions of un-annotated genes more accurately.

Chapter 6 describes how the discovered interesting structural patterns can be applied on multiple attribute biological networks for characterization and discrimination among them. By analyzing the relationships between proteins, we can discover a set of interesting patterns that characterize each particular class in terms of interestingness. By defining such interesting subgraphs as feature vectors, we show how a classification model that is based on the total weight-of-evidence can be discovered to discriminate an unknown sub-network among different classes.

Chapter 7 introduces how the proposed algorithm is useful in discovering class-specific patterns from molecular data. The molecular structures store the information that defines the functions of their physical and chemical properties, and we describe in this chapter how we can make use of a novel algorithm to identify patterns that characterize each class of molecular data based on the interestingness measure.

Chapter 8 describes how the interestingness measure we proposed here can be applied to identify protein complexes in PPI networks. A PPI network contains not only the information of individual interactions between protein-pairs, more

biological meaningful pattern such as protein complexes are also included. We describe in this chapter how we can combine the biological and structural information in the PPI networks to capture the significant relationships between proteins, and define a significant score for each interaction for identifying interesting subgraphs as protein complexes.

Finally, in Chapter 9, the contributions of the work proposed here are summarized and potential improvements and future extension for this dissertation are discussed.

Chapter 2

Background and Related Work

This chapter describes the basic concepts of biological interaction network, and provides a review of the literatures related to the structural analysis of biological interaction networks. The analysis of complex interaction networks has become an important task in studying molecular biology. Various network theories are developed for a systematic characterization on the network topology.

2.1 Biological interaction network

Biological networks involve the interactions between basic biological units like genes and proteins. A gene is an essential part of the DNA and genes encode proteins which are functional building blocks for cells.

2.1.1 Topology of biological network

Among the different biological networks that have been looked at, protein-protein interaction (PPI) networks and gene regulatory networks (GRNs) have received the most attention. A PPI network can be a very complex biological network as the cell of an organism may contain thousands of functional modules. A group of functionally related proteins is tightly connected as a protein module, while proteins

from different functional units are more loosely connected [57]. Most highly connected proteins in the cell are the most important for its survival [66].

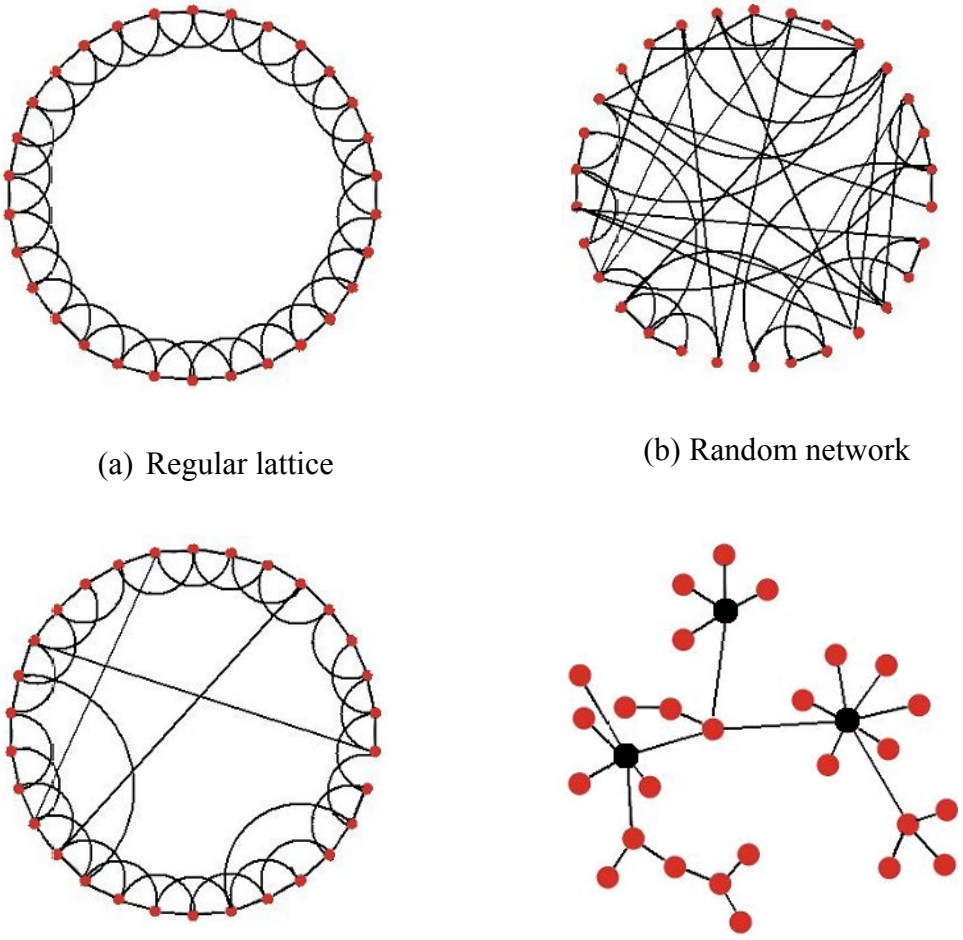
Gene regulatory network is another kind of biological network that is widely studied. It is a collection of DNA segments (genes) in a cell which interact with each other and with other substances in the cell. As one gene can affect the expression of another gene by binding of the gene product (protein) of one gene to the promoter region of another gene, gene regulatory network is used to model the interactions between genes and proteins, such as transcription factors, with the level of gene expression.

To perform effective network analysis, it is important to understand the fundamental properties of biological networks. The general structural principles of biological networks are regularity (nearest neighbors are more likely to be directly connected) and randomness (randomly interconnected vertices). However, most biological systems are neither regular nor random. The biological networks have been regarded as exhibiting small-world network characteristics [58] by Watts and Strogatz [59]. In a small-world network, most vertices are not neighbors of one another, but they can be reached from every other by a small number of steps. It captures the small world phenomenon of strangers being linked by mutual acquaintances. The case is similar for the biological network such as PPI network, where proteins that share a number of neighbors are more likely to have a function in common. However, the significance of two proteins sharing a particular number of neighbors is dependent on the number of neighbors that each has.

Another property of biological network is its scale-free nature [60]. The most notable characteristic in a scale-free network is the relative commonness of vertices with a degree that greatly deviate from the random distribution. The degree distribution of a scale-free network follows a power-law distribution [7] that refers to the number of vertices with degree k and exponent constant r that is greater than 0. The power-law degree distribution would be much more likely to have vertices with a very high degree. The homologous gene and protein sequences are conversed across species, similarly, the common interaction patterns that represent essential functions are also expected to be retained in biological networks [61]. Currently available protein-protein interactions cover only a fraction of the complete PPI networks. These partial networks display scale-free topologies. The discovered interaction pattern in biological network is often referred to as a module [62] that contains a group of interacting biomolecules.

The topologies of four basic types of network model, regular lattice, random, small-world, and scale-free network, are shown in *Figure* 1 [85]. Regular lattice (*Figure* 1a) represents the simplest type of network. The whole network density is low and the nodes are usually linked to its immediate neighbors. In a random network (*Figure* 1b), the placement of links is random, and most nodes will have approximately the same number of links. Biological networks are neither one of these types, and it is more likely to belong to the small-world network and scale-free network. A small-world network (*Figure* 1c) contains sets of nodes connected with mostly immediate connections and a few randomly connected long distance connections. Its degree of randomness is lying between the regular lattice and the

random network. A scale-free network (*Figure* 1d) is created by preferential attachment, in which a node is linked to those that already have many links [83]. Hence, most nodes have only a few links and they are held together by a few highly connected "hubs" (black nodes). This imbalance between the sparsely connected nodes and "hubs" plays an important role in deriving the functionality of the network [84].



(a) Regular lattice

(b) Random network

(c) Small world network

(d) Scale free network

Figure 1 The topologies of four basic network models

2.1.2 Availability of biological data

Due to advances in technologies to obtain complete genome sequences and high-throughput post-genomic experimental data, more and more public repositories are made available online to facilitate the analysis. The currently available biological networks can be derived from scientific literature corpus and experimentally identified at cellular level. Different methods for obtaining such networks can result in significantly different networks, for example, PPI network is determined by physical methods like yeast two-hybrid system and affinity purification coupled to mass spectrometry, and metabolic networks are determined through biochemical experiments. Although abundant data are available on the web, each data source is characterized by its own data structure and query interface.

PubMed [86] repository stores scientific papers as semi-structured data in XML format. It is maintained by the U.S. National Institutes of Health (NIH) and it provides free digital archive of biomedical and life sciences journal literature.

The Gene Ontology project [8] uses ontologies to store structured controlled vocabularies that describe gene products in terms of their associated biological processes, cellular components, and molecular functions. An ontology provides the representation of a set of concepts and the relationships among them. They provide a high level abstraction of data which may facilitate both expressing of complex queries over a single source, and querying of several heterogeneous sources by exploiting a common set of concepts.

Other repositories store data in relational databases in order to represent the relations between the various instances of different classes. Relational databases provide a robust and query-efficient technology to store data. However, because of its limited complex-modeling capabilities, relational modeling may lack the flexibility needed to represent complex data types. Relational-based repositories typically store a single information type and rely on the query engine of the database management system for providing form-based data access. Some popular public repositories of the relational databse type are listed in *Table* 1.

Table 1 Public repositories of biological network

Database	**# of interactions (# of organisms)**	**Types**[1]	**Website**
MIPS [14]	15,488 (10)	P	http://mips.gsf.de/proj/ppi
DIP [15]	57,683 (274)	P	http://dip.doe-mbi.ucla.edu
BioGRID [16]	240,207 (22)	P, G	http://www.thebiogrid.org
STRING [17]	730,000 (630)	P	http://string.embl.de
MINT [68]	111,437 (30)	P	http://mint.bio.uniroma2.it/mint
IntAct [49]	194,558 (9)	P	http://www.ebi.ac.uk/intact
HPRD [76]	38,806 (470)	P	http://www.hprd.org
KEGG [70]	96,232 (1,015)	M	http://www.genome.jp/kegg
MetaCyc [77]	7,837 (1,735)	M	http://metacyc.org
Reactome [78]	26,216 (23)	M	http://www.reactome.org
GIN [79]	54.554 (8)	G	http://gin.ncibi.org
GeneNet [80]	3,634 (93)	G	http://wwwmgs.bionet.nsc.ru
ITFP [81]	124,591 (3)	G	http://itfp.biosino.org/itfp

[1] P: PPI network, M: Molecular network, G: gene regulatory network

These public repositories are distributed across heterogeneous platforms. In order to provide an integrated view of heterogeneous databases, data warehouses are used to store data from different sources. These data are consolidated into a single local repository. For example, *TAMBIS* [88] is a mediator-based and ontology-driven integration system developed for such purpose. Its queries are formulated through a graphical interface where a user browses through concepts defined in a global schema and selects the relevant ones for the particular query. *Biozon* [87] is another recent data warehousing project that aims to find ways to store data obtained from various heterogeneous biological sources. Several types of derived data such as similarity relationship between biomolecules and functional predictions are also stored there. This derived data are obtained by expanding on existing data types based on refining of existing objects. New data types can then be obtained by processing existing and derived data. Biozen employs a vertical integration approach, where sources are not only incorporated into a single schema but are also integrated using a non-redundant object-centric model. Borrowing the concepts from these integration systems, we also extract information from heterogeneous data sources and store them in a central data warehouse (*Figure* 2) for further query.

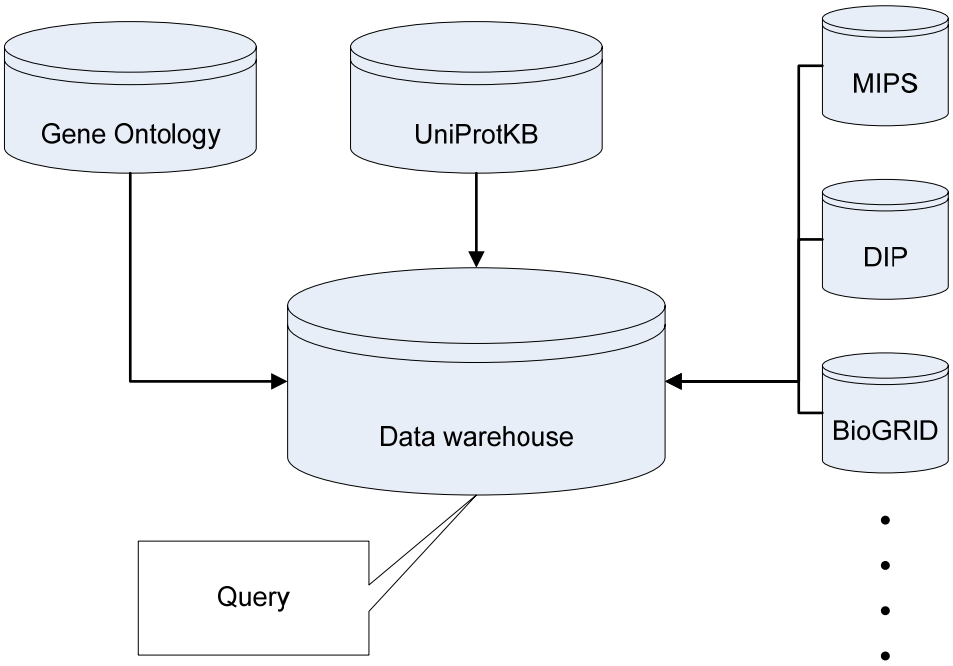


Figure 2 Data Integration of biological databases

2.2 Structural analysis of biological networks

In order to discover structural patterns between biological interaction networks, network alignment is one of the more popular approaches that have been used to identify conserved patterns in biological networks.

Von Mering et al. [65] were among the earliest to investigate into the problem of inferring protein interactions from high-throughput data. They had built a database [17] of functionally associated protein pairs derived from computational integration of direct biomolecules. The method they used to find such protein pairs is beased on the intersection of direct experiments. The method is not considered very effective as it achieved a relatively low false positive rate with low coverage. Later,

Jansen et al. [82] proposed to predict the function of biomolecules with various biological features that include biological function data from the Gene Ontology [8], the MIPS functional catalog [14], and the high-throughput experimental interaction data, etc. Based on this same idea, Lin et al. [73] attempted to quantify functional similarity between two proteins. They considered the direct interactions between proteins when inferring the presence of biological networks. Due to the problem of noise and redundancies in the network, the direct interaction approach is not robust enough to identify interacting associations. Instead of using the direct neighbor as a scoring function, Samanta et al. [47] considered the neighborhood interaction pairs to define the network graph, and employed hypergeometric p-value to define the significance relations.

The identification of association patterns between biomolecules provides useful insights for the understanding of the topologies of biological networks. The topology of a network refers to the relative connectivity of its nodes. Different topologies affect specific network properties and it is important to understand and model the topological and dynamical properties of various biological networks in a quantifiable manner. There have been approaches to discover frequently occurring substructures in biological networks as a way to model them. A number of previous studies [23], [40], and [42] pointed out the existence of network motifs in PPI graphs and transcription regulation networks. Many complex networks have been shown to have certain structural design principles. The convergent evolution towards the same motif types has also been seen in the transcription-regulatory network of diverse species. All these observations further indicate that motifs are indeed of direct

biological relevance. However, it should be noted that not all motifs in a biological network are equally significant or important in real biological networks.

Biological networks are composed of interacting modules with different functions [32, 33, 34], and these functional modules, such as protein complexes in PPI networks, are connected subgraphs that are expected to repeat in the networks among the same or different species. Many studies [35, 36, 37, 38] are therefore focused on discovering network motifs using network topologies. Hartwell et al. [25] suggested that understanding the relatedness of the modules is useful because knowledge about one member of a class can inform the study of the others, and the modular structure can further facilitate the understanding of cellular organization, transcription regulation and phenotypic evolution, etc. It is believed that network motifs are useful to describe the structural organization and functionality of a biological network. As a biological network can be naturally represented as an undirected graph such that each vertex represents a biomolecule and each edge represents the interaction between two vertices, graph mining algorithms such as network alignment and frequent subgraph mining approach are developed for the task of motif discovery.

An early study of biological network alignment is introduced by Ogata el al. [96] to find functionally related enzyme clusters in metabolic networks using a simple heuristic of global alignment. Matthews et al. [26] apply similar ideas to identify homologous pairs of interactions from two PPI networks. Later, Kelley et al. [27] propose a PATHBLAST algorithm to align two PPI networks and combine

them into a global alignment graph. An alignment graph is constructed with vertices representing pairs of homologous modules and edges representing the type of conserved interactions using similarity measurements like BLAST [28] that compares primary biological sequence information. This work has been extended by Sharan et al. [6] by introducing a probabilistic model and heuristic greedy approach to search for dense subgraphs in the global alignment graph of yeast and bacteria, and this is then modified to align three networks simultaneously in [29]. However, all these approaches that aim at computing either local or global alignments involves many-to-many similarity measurements between module pairs that causes the alignment graph to grow exponentially with the number of aligned networks. When the number of networks is increased, the computation complexity is also increased exponentially. To simplify the problem and improve the efficiency, the identification of functional modules is emerged to discover common sub-networks by graph mining approaches.

A number of graph mining algorithms have been developed to discover potentially useful subgraphs in data with complex structures and relationships. Dehaspe et al., for example, proposed such an ILP-based graph mining algorithm [74] called WARMR [75] that searches for frequent subgraphs in a graph database using first order predicate logic to represent graph data. ILP-based graph mining algorithms have some limitations in that they may not be robust enough to deal with noisy biological data. Under such circumstances, the computational complexity of such algorithm can be too high to afford as it performs a lot of tests for equivalence in order to prune infrequent and semantically redundant queries. Other than the ILP-

based approach, there are quite a number of other graph mining algorithms that can be used to discover frequent subgraphs in structural data, *FSG* and *gSpan* are such examples.

FSG [40] adopts an edge-based candidate generation strategy that expands on a subgraph based on a level-by-level expansion approach very much like that of the Apriori algorithm [39]. *FSG* begins its work by enumerating all frequent single and double-edge subgraphs. It then generates larger candidate subgraph iteratively with each iteration generating subgraphs that have one more edge than those generated in the previous iteration. For *FSG* to function well, it has to rely on canonical labeling to check whether a particular subgraph satisfies a support threshold. Thus, if two graphs are isomorphic with each other, their canonical labels must be identical. Unfortunately, this process of canonical labelling and determining of graph isomorphism is memory consuming for large databases!

gSpan [45] is another subgraph mining algorithm that discovers frequent subgraphs based on graph canonical forms and it searches for frequent subgraphs by depth-first search (DFS). It does so by starting from a randomly chosen vertex. It then visits vertices and marks them with their status. The set of vertices visited is expanded repeatedly until a full depth-first search tree is built. One graph may have different trees built with DFS depending on the order in which the vertices were visited. *gSpan* discovers all frequent subgraphs without generating candidates or pruning false positives.

Other than the graph mining algorithms, *WARMR*, *FSG*, and *gSpan*, another algorithm for finding frequent subgraphs is called *Gaston* [114]. It discovers such patterns by first finding frequent paths, then trees, and then cyclic graphs. It stores all occurrences of these graphs in an embedding list so that the frequency of occurrence of a subgraph can be determined by scanning the embedding list without searching in the original graph. *Gaston* speeds up the enumeration and discovery process by so doing. The mining performance of *Gaston* in terms of running speed and memory consumption are compared with the other graph mining algorithms and reported in [115].

MoFa [116] also finds frequent subgraphs in molecular network data. It does so by maintaining parallel embeddings for both vertices and edges throughout the graph mining process. Like *Gaston*, each such embedding consists of a set of references to a molecule that point to the atoms and bonds that form a subgraph. Such embeddings can be extended so that larger subgraphs can be formed iteratively [116].

Subdue [23] is another graph mining algorithm that discovers frequent subgraphs but it is quite different from the above as it makes use of the minimum description length principle to narrow down possible outcomes when trying to identify subgraphs that best compress the original graph. Unlike *FSG*, *gSpan*, *Gaston* and *MoFa*, etc., the subgraphs that *Subdue* discovers only refer to the abstract patterns defined in terms of previously discovered patterns. As a result, some interesting patterns that occurred frequently could be missed.

All of the above graph mining algorithms can discover a set of frequent subgraphs from biological network successfully. From the discovered subgraphs, we can observe that some patterns are repeated in the network as motifs and the distribution of the motifs characterizes the local structure of a network. *Figure* 3 shows an example of frequent subgraph enumeration where (a) to (c) are a set of attributed graphs and (d) to (i) are the frequent subgraphs with support threshold equals to 60%.

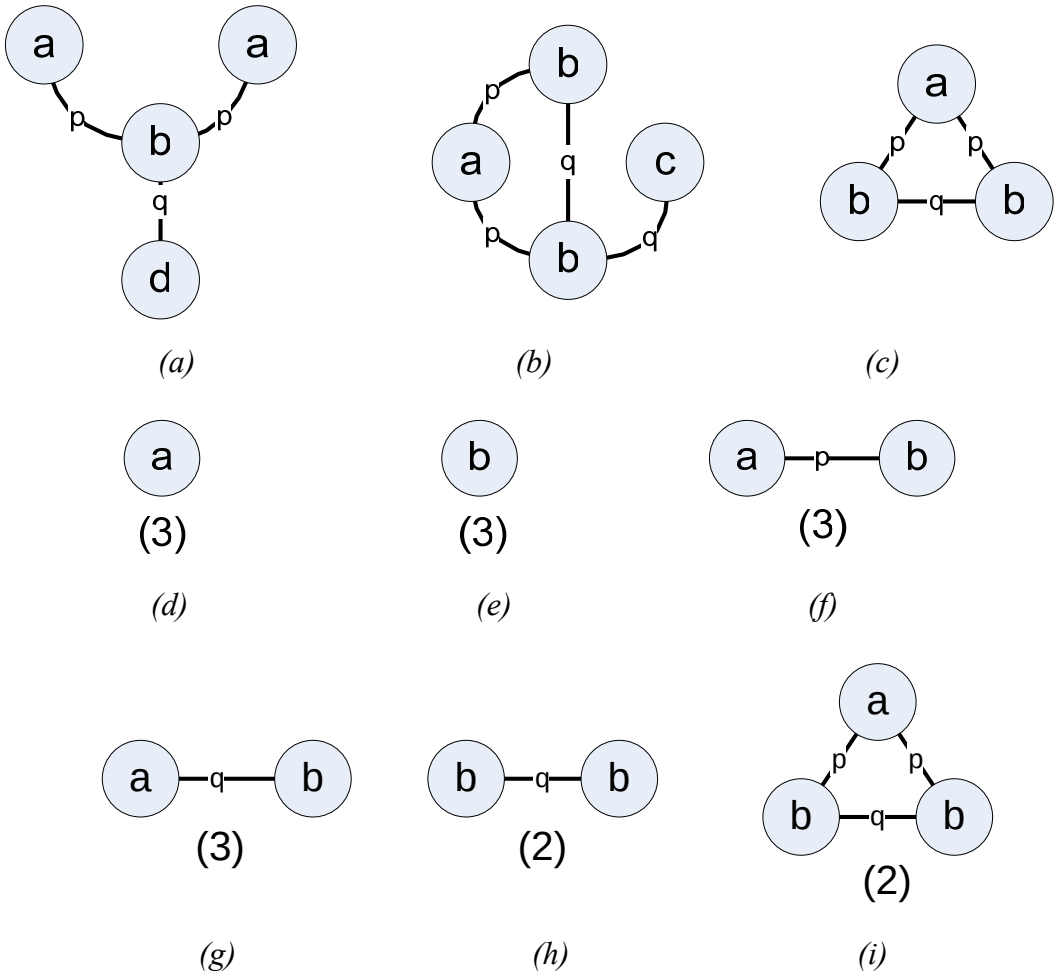


Figure 3 (a-c) A set of attributed graphs (d-i) frequent subgraphs with support threshold 60%

They discover frequent subgraphs by building on smaller subgraphs edge-by-edge to capture frequently-occurring patterns in each class of graphs, and their kernels are subgraph isomorphism which is known to be NP-complete. However, these subgraphs are only quantified by the frequency, and the frequency level has no additional meaning other than showing the number of occurrences. These algorithms try to identify frequent subgraphs against a user-defined threshold. If the threshold is set too small, one may not be able to discover enough frequent subgraphs to allow classes to be distinguished from each other. If the threshold is set too large, one may discover too many frequent subgraphs and many of them may be irrelevant to the classification process. As subgraphs that appear frequently in one graph can also appear frequently in other classes, they may not be very useful for graph classification when the classes are similar.

A different concept should be introduced to discover interesting motifs that can represent different levels of network organization and provide true characterization among different networks. We have to develop a way to identify the interestingness of a subgraph in a class so that it can be distinguishable among different classes for classification purpose.

Chapter 3
Filtering Out Uninteresting Structural Patterns

In this chapter, we describe our proposed approach to define an interestingness measure for solving the problem of identifying interesting structural patterns from biological structural data. Given a set of biological structural data that is pre-classified into a number of classes, a number of graph mining algorithms have been applied to discover useful subgraphs in the data so that "unseen" structures not originally in the data set can be accurately classified. This is done typically by first representing the structural data in graphs and then using graph mining algorithms to discover frequently occurring subgraphs in them. The molecular substructures that these frequently occurring subgraphs correspond to are considered interesting. Such an approach has been shown to be effective in some cases. However, in other cases, a substructure that occurs frequently in one class may also does so in another. The discovering of frequent subgraphs in molecular graphs may therefore not always be the most effective approach for classification.

By making use of a test statistic, we propose *MISPAG-FP* to screen each frequent subgraph discovered and determine if they are interesting. The degree of interestingness of these subgraphs are then determined using an information-theoretic measure.

3.1 Frequent Subgraph Discovery

Biological networks enable the characterization of biological processes and the determination of useful substructures in such network structure is important to understand such biological processes as cellular organization, transcription regulation and phenotypic evolution. To discover frequent subgraphs in a graph database, there are several graph mining algorithms to choose from.

Given a set of graphs, these algorithms can be used to mine frequent subgraphs in them. Recently, these algorithms have also been used to for graph classification [117, 50]. The most popular among these algorithms are *FSG* [40] and *gSpan* [45]. With the discovered frequent subgraphs discovered by these algorihtms, we propose to use *MISPAG-FP* to screen out frequent subgraphs that are irrelevant and retaining those that are useful for the characterization of molecular classes and the discrimination of one class from another. *MISPAG-FP* can be shown to be able to increase classification accuracy.

Given a set of graph data represented as a set of graphs, $\mathfrak{G} = \{G_1, \ldots, G_j, \ldots, G_N\}$, one can use either of these algorithms to discover a set of frequent subgraphs, $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(p)}, \ldots, \mathbf{S}^{(P)}$, where $\mathbf{S}^{(p)} = \{ S_1^{(p)}, \ldots, S_s^{(p)}, \ldots, S_{m_p}^{(p)} \}$, $p = 1, \ldots, P$, for each of the corresponding p classes, $\mathfrak{C}^{(1)}, \ldots, \mathfrak{C}^{(p)}, \ldots, \mathfrak{C}^{(P)}$.

3.1.1 The *FSG* Algorithm

The *FSG* algorithm can find all frequent subgraphs in each class of molecular graphs using the Apriori Algorithm [39]. It does so by treating edges in the graphs as items in transactions so that the Apriori Algorithm can be used to discover frequent subgraphs like it is used to discover frequent itemsets, i.e., in the same way the Apriori Algorithm increases the size of frequent itemsets by adding a single item at a time, the *FSG* algorithm also increases the size of frequent subgraphs by adding an edge one by one.

Briefly, the *FSG* can be described as follows. For each $\mathfrak{G}^{(p)}$, $p = 1, \ldots, P$, *FSG* first finds a set of frequent one-edge subgraphs and a set of frequent two-edge subgraphs. Then, based on these two sets of intermediate subgraphs, it starts to iteratively generate candidate subgraphs whose size is greater than the previous frequent subgraphs by one edge. *FSG* then counts the frequency for each of these candidates and prunes subgraphs that do not satisfy the support threshold σ. The qualified subgraphs are further expanded and their frequencies are verified with the same support condition to prune the lattice of frequent subgraphs. The final set of frequent subgraphs $\mathbf{S}^{(1)}$, …, $\mathbf{S}^{(p)}$, …, $\mathbf{S}^{(P)}$, where $\mathbf{S}^{(p)}$ contains all frequent k-subgraphs is generated for each class. Let g^k be a k-subgraph with k edges, $\mathcal{D}^k$ be a set of candidate subgraphs with k edges, $\mathbf{S}^{k(p)}$ be a set of frequent k-subgraphs for class $\mathfrak{G}^{(p)}$, the algorithm of *FSG* can be summarized in *Figure* 4.

Algorithm of *FSG* [40**]**

Input : Input graphs $\mathfrak{G}$, support threshold σ

Output : A set of frequent subgraphs $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(p)}, \ldots, \mathbf{S}^{(P)}$

for each $\mathcal{G}^{(p)} \in \mathfrak{G}$ **do**

$\mathbf{S}^{1\,(p)}$ = all intermediate frequent 1-edge subgraphs in $\mathcal{G}^{(p)}$;

$\mathbf{S}^{2\,(p)}$ = all intermediate frequent 2-edge subgraphs in $\mathcal{G}^{(p)}$;

$k = 3$;

$\mathbf{S}^{(p)} = \mathbf{S}^{1\,(p)} \cup \mathbf{S}^{2\,(p)}$

while $\mathbf{S}^{k-1\,(p)}$ is not null **do**

$\mathcal{D}^k$ = candidate generated from $\mathbf{S}^{k-1\,(p)}$

for each candidate g^k in $\mathcal{D}^k$ **do**

Initialize the count of g^k

for each graph transaction $G \in \mathcal{G}^{(i)}$ **do**

if G contains g^k

Increment the count of g^k by 1

for each candidate g^k in $\mathcal{D}^k$ **do**

if count of g^k is greater than or equal to σ

$\mathbf{S}^{k\,(p)} = g^k$

$\mathbf{S}^{(p)} = \mathbf{S}^{(p)} \cup \mathbf{S}^{k\,(p)}$

Increment k by 1;

return $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(p)}, \ldots, \mathbf{S}^{(P)}$

Figure 4. Algorithm of FSG

3.1.2 The *gSpan* Algorithm

The *gSpan* algorithm [45] discovers a set of frequent subgraphs for each graph class by mapping each graph in the class to a unique minimum DFS code as the canonical label. Firstly, *gSpan* sorts all vertices and edges in the set of graph transactions in each class according to their frequency of occurrence and removes the infrequent vertices and edges from $\mathfrak{G}^{(p)}$. The remaining vertices and edges are relabeled and sorted in descending frequency. $\mathbf{S}^{1(p)}$ is then formed by all frequent one-edge subgraphs and it acts as the seed for generating more children. The sub-procedure, called *SubgraphMiner* expand each one-edge frequent subgraph $\mathbf{S}^{1(p)}$ from each class by adding one edge at a time. In the *SubgraphMiner*, if s is the minimum DFS code of the graph it represents, it adds s to its frequent subgraph set $\mathbf{S}^{(p)}$. It then generates all potential children with a one-edge growth and runs SubgraphMiner recursively for each child. After that, the edge is removed from each graph in $\mathfrak{G}^{(p)}$ after all descendants of this one-edge graph have been searched. When all frequent k-subgraphs and their descendants are generated, the final set of frequent subgraphs $\mathbf{S}^{(p)}$, $p = 1, \ldots, P$. will be generated for each class. The algorithm of *gSpan* can be summarized in *Figure* 5.

Algorithm of *gSpan* [45**]**

Input : Input graphs $\mathfrak{G}$, support threshold σ

Output : A set of frequent subgraphs $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(p)}, \ldots, \mathbf{S}^{(P)}$

for each $\mathfrak{G}^{(p)} \in \mathfrak{G}$ **do**

Sort the labels of all vertices and edges in $\mathfrak{G}^{(p)}$ and by frequency

Remove infrequent vertices and edges

Re-label and sort the remaining vertices and edges

$\mathbf{S}^{1(p)}$ = all frequent 1-edge subgraphs in $\mathfrak{G}^{(p)}$;

$\mathbf{S}^{(p)} = \mathbf{S}^{1(p)}$ that sorted in DFS lexicographic order

for each edge in $\mathbf{S}^{1(p)}$ **do**

Initialize *s* with *e*

SubgraphMiner($\mathfrak{G}^{(p)}$, $\mathbf{S}^{(p)}$, *s*)

Remove *e* from $\mathfrak{G}^{(p)}$

if $|\mathfrak{G}^{(p)}|$ is less than σ

break

return $\mathbf{S}^{(1)}$, …, $\mathbf{S}^{(p)}$, …,$\mathbf{S}^{(P)}$

SubgraphMiner

Input : $\mathfrak{G}^{(p)}$, $\mathbf{S}^{(p)}$, *s*

if *s* is not the minimum DFS code

return

Find all embeddings of *s* in $\mathfrak{G}^{(p)}$ and add to $\mathbf{S}^{(p)}$

Generate all potential children *c* of *s* in $\mathbf{S}^{(p)}$ with one edge growth

for each *c* **do**

if support of *c* is greater than σ

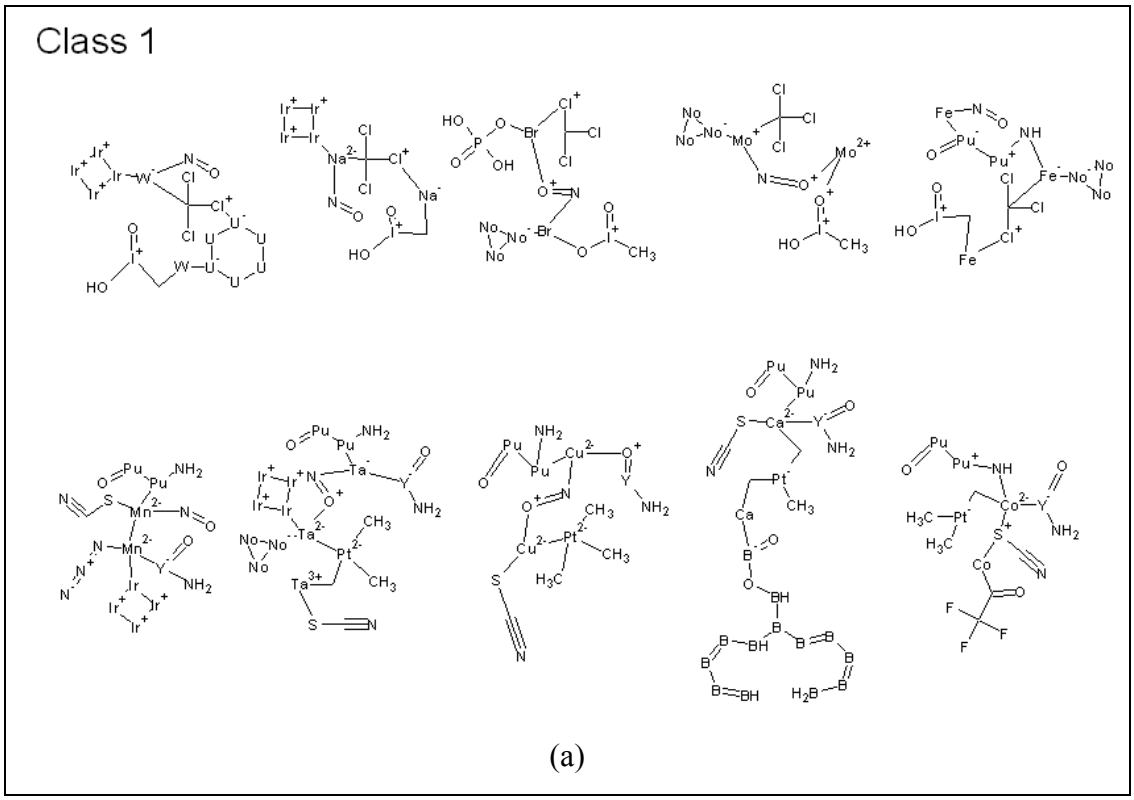
Add *c* to *s*

SubgraphMiner($\mathfrak{G}^{(p)}$, $\mathbf{S}^{(p)}$, *s*)

Figure 5. Algorithm of gSpan

3.2 Illustrative Example

To explain why the discovering of frequent subgraphs may not always be useful for graph classification, let us consider an example. We are given three classes of artificial molecular data shown in *Figure* 6. Each of these three classes of data contains ten molecules and each molecule consists of atoms connected with bonds. These molecules are generated in such a way that the atoms are chosen from 30 possible atoms, including such atoms as carbon (C), oxygen (O), iridium (Ir), nobelium (No) and thorium (Th), and bond types from three possible types including single, double and triple bonds. These molecules can be represented as labeled molecular graphs with each node used to represent an atom and each edge a bond.



(a)

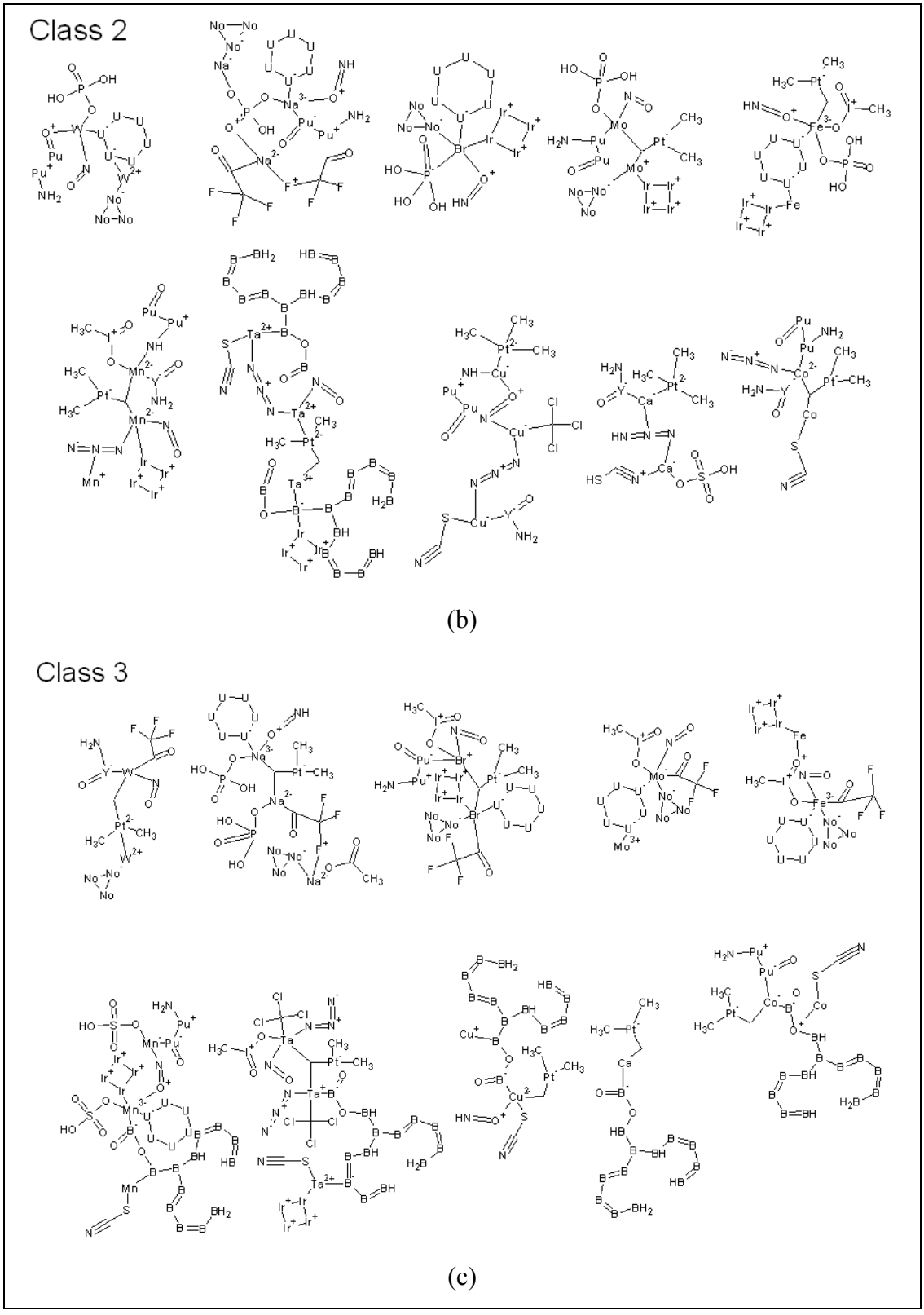


Figure 6. Training molecular data

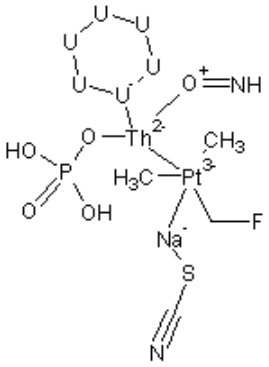


Figure 7. Testing sample

Given the set of graph data as shown in *Figure* 6, frequent subgraphs can be discovered in each of *Class* 1, 2 and 3 using a graph mining algorithm such as *FSG* and *gSpan*. These algorithms require that a threshold be given by the users to define how frequent a subgraph should appear for it to be considered frequent. For the purpose of illustration, we choose *FSG*. This is because, even though the support thresholds are set the same, the results obtained by *gSpan* can be different from that of *FSG* as *gSpan* does not perform subgraph pruning. *FSG*, on the other hand, can discover maximal frequent subgraphs and can better avoid the problems caused by the discovering of subgraphs that are too fragmented.

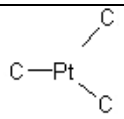
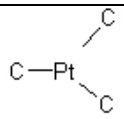
By setting a support threshold of 80% (i.e. any subgraph that occurs in at least eight out of ten graphs), the frequent subgraphs that are found in each of the three classes of graphs are listed in *Table* 2. It should be noted that the same frequent subgraph, a nitrogen atom double-bonded with an oxygen atom (i.e. N═O), appears in 80% of the graphs in each of the three classes.

Table 2. Maximal Frequent subgraphs (support threshold = 80%)

***Class* 1**		**Frequency**	***Class* 2**		**Frequency**	***Class* 3**		**Frequency**
$S_1^{(1)}$	N═O	8	$S_1^{(2)}$	N═O	8	$S_1^{(3)}$	N═O	8

Since the choice of threshold does not allow any unique frequent subgraph to be discovered for each class, we lower the support threshold by 10%. The results are shown in *Table* 3. More frequent subgraphs are discovered this time when the support threshold is lowered to 70%. However, the newly discovered frequent subgraphs for *Class* 2 and 3 are still the same and a graph with such subgraphs may be classified into either *Class* 2 or 3. This means that the discovered frequent subgraph cannot allow graphs in *Class* 2 to be easily discriminated from *Class* 3.

Table 3. Maximal Frequent subgraphs (support threshold = 70%)

***Class* 1**	**Frequency**	***Class* 2**	**Frequency**	***Class* 3**	**Frequency**
N═O	8	N═O	8	N═O	8
			7		7

When the support threshold is further lowered to 60%, more frequent subgraphs are discovered and they are shown in *Table* 4. Unfortunately, the newly discovered frequent subgraphs for each of the three classes still overlap with each other. A graph characterized by these subgraphs can be classified into one or more classes. For example, if a graph G is characterized by the subgraph  , it can be classified

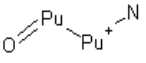
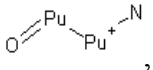
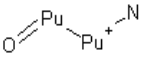
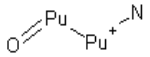
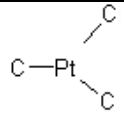
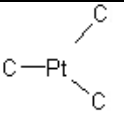
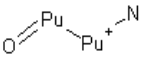
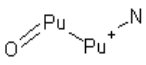
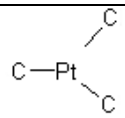
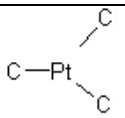
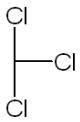
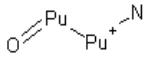
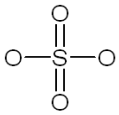
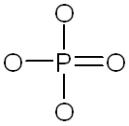
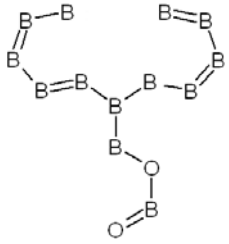
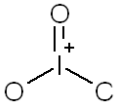
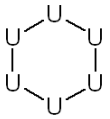
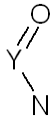
into either *Class* 2 or 3. If G is characterized by the subgraph  , it can be classified into either *Class* 1 or 2. If G is characterized by both  and  , then there is a chance that it can be classified into any of *Class* 1, 2 or 3 as  appears 6 times in *Class* 1 and 2, and  appears 7 times in *Class* 2 and 3.

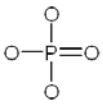
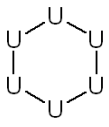
Table 4. Maximal frequent subgraphs (support threshold = 60%)

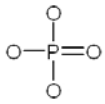
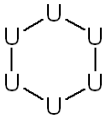
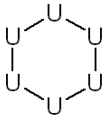
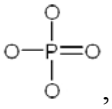
***Class* 1**		**Freq.**	***Class* 2**		**Freq.**	***Class* 3**		**Freq.**
$S_1^{(1)}$	N═O	8	$S_1^{(2)}$	N═O	8	$S_1^{(3)}$	N═O	8
$S_2^{(1)}$		6	$S_2^{(2)}$		7	$S_2^{(3)}$		7
			$S_3^{(2)}$		6			

To find more interesting and useful frequent subgraphs for classification, the support threshold is further lowered to 50%. Using the *FSG* again, the frequent subgraphs discovered are shown in *Table* 5. This time, many more frequent subgraphs are discovered and some of the subgraphs discovered in each of $\mathbf{S}^{(1)}$, $\mathbf{S}^{(2)}$ and $\mathbf{S}^{(3)}$ do not overlap with each other.

Table 5. Maximal frequent subgraphs (support threshold = 50%)

***Class* 1**		**Freq.**	***Class* 2**		**Freq.**	***Class* 3**		**Freq.**
$S_1^{(1)}$		8	$S_1^{(2)}$		8	$S_1^{(3)}$		8
$S_2^{(1)}$		6	$S_2^{(2)}$		7	$S_2^{(3)}$		7
$S_3^{(1)}$		5	$S_3^{(2)}$		6	$S_3^{(3)}$		5
$S_4^{(1)}$		5	$S_4^{(2)}$		5	$S_4^{(3)}$		5
$S_5^{(1)}$		5	$S_5^{(2)}$		5	$S_5^{(3)}$		5
$S_6^{(1)}$		5	$S_6^{(2)}$		5	$S_6^{(3)}$		5
$S_7^{(1)}$		5						

If we are to classify the testing sample in *Figure* 7, it should be noted that this graph is characterized by three frequent subgraphs ,  and  from each of $\mathbf{S}^{(1)}$, $\mathbf{S}^{(2)}$ and $\mathbf{S}^{(3)}$ respectively as shown in *Figure* 8. It is therefore hard to decide which class this graph should be classified into based on

these subgraphs that it contains. If one is to take a closer look at the frequency of appearance of each of these three subgraphs, ,  and  , in each class, one may discover that even though is not frequent enough in *Class* 2 and 3, it appears in 40% of the graphs in these classes. This is the case also with  . Although it only appears in 10% of the graphs in *Class* 1, it appears in 40% of the graphs in *Class* 2. Of these three subgraphs,  , is the most interesting and unique in the sense that, while it appears in 50% of the graphs in *Class* 2, it only appears in 10% of the graphs in both *Class* 1 and 3. In other words, this subgraph provides more evidence for a graph it characterizes to be classified into *Class* 2 than other subgraphs. In fact, it is for this reason that the graph in *Figure* 7 belongs more likely to *Class* 2 than any other classes.

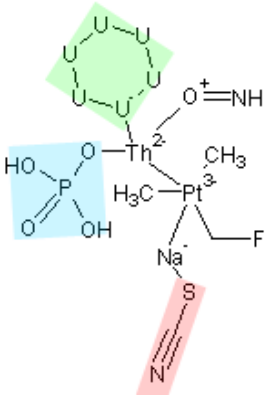
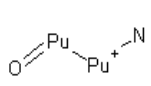
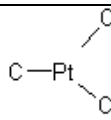
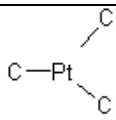
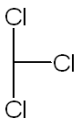
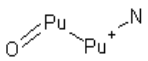
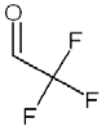
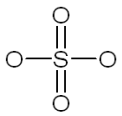
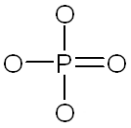
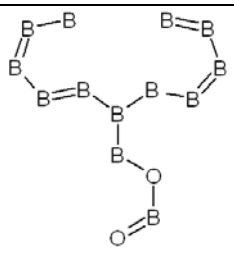
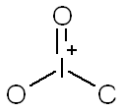
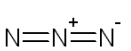
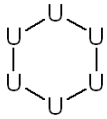
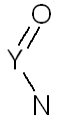
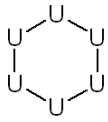
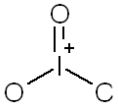
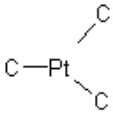
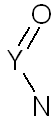
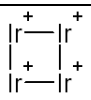


Figure 8. Classifying the unseen molecule in Figure 2 with FSG

In order to discover more frequent subgraphs that may be useful for classifying the unseen molecule, the support threshold is further reduced to 40%, and the new frequent subgraphs are discovered as shown in *Table* 6. The newly discovered subgraphs are $S_8^{(1)}$, $S_9^{(1)}$, $S_{10}^{(1)}$ in *Class* 1; $S_7^{(2)}$, $S_8^{(2)}$, $S_9^{(2)}$, $S_{10}^{(2)}$ in *Class* 2 and $S_7^{(3)}$, $S_8^{(3)}$, $S_9^{(3)}$ in *Class* 3. Although the support threshold is lowered to 40%, these subgraphs are all appeared more frequently in the other classes, for example, $S_8^{(1)}$ is previously discovered as frequent subgraph $S_2^{(2)}$ in class 2 and $S_2^{(3)}$ in class 3. The case is the same as the others. We tried to further reduce the support threshold to 30%, but the case is still the same that the newly discovered subgraph is already found at higher threshold value.

Table 6. Maximal frequent subgraphs (support threshold = 40%)

***Class* 1**		**Freq.**	***Class* 2**		**Freq.**	***Class* 3**		**Freq.**
$S_1^{(1)}$	N═O	8	$S_1^{(2)}$	N═O	8	$S_1^{(3)}$	N═O	8
$S_2^{(1)}$		6	$S_2^{(2)}$		7	$S_2^{(3)}$		7
$S_3^{(1)}$		5	$S_3^{(2)}$		6	$S_3^{(3)}$		5
$S_4^{(1)}$		5	$S_4^{(2)}$		5	$S_4^{(3)}$		5

$S_5^{(1)}$		5	$S_5^{(2)}$		5	$S_5^{(3)}$		5
$S_6^{(1)}$		5	$S_6^{(2)}$		5	$S_6^{(3)}$		5
$S_7^{(1)}$		5	$S_7^{(2)}$		4	$S_7^{(3)}$		4
$S_8^{(1)}$		4	$S_8^{(2)}$		4	$S_8^{(3)}$		4
$S_9^{(1)}$		4	$S_9^{(2)}$		4	$S_9^{(3)}$		4
$S_{10}^{(1)}$		4	$S_{10}^{(2)}$		4			

The actual relative frequency of appearances of each frequent subgraph in each class may therefore provide useful information for classification. The idea that *MISPAG-FP* uses to filter out uninteresting and irrelevant frequent subgraphs to allow molecular classification to be performed effectively is therefore to take into consideration such information so as to measure the relatively interestingness of each frequent subgraph relative to the others.

3.3 Interestingness Measure with Adjusted Residual Analysis

Here we present an interestingness measure of *MISPAG* to identify interesting subgraphs that are interesting and useful for classification. This methodology is based on the use of an adjusted residual analysis [12, 13].

Once the set of frequent subgraphs, $\mathbf{S}^{(p)}$, are discovered for each of $\mathfrak{G}^{(p)}$, p = 1, …, P, respectively, the probability that a graph, G, is in $\mathfrak{G}^{(p)}$, $p \in \{1, \ldots, P\}$ given that G is characterized by a frequent subgraph, $S_j^{(p)} \in \mathbf{S}^{(p)}$, $j \in \{1, \ldots, m_p\}$ can be determined as

$$\Pr(G \in \mathfrak{G}^{(p)} \mid G \text{ is characterized by } S_j^{(p)})$$

$$= \frac{\text{total no. of graphs in } \mathfrak{G}^{(p)} \text{ that are characterized by } S_j^{(p)}}{\text{total no.of graphs in } \mathfrak{G} \text{ that are characterized by } S_j^{(p)}} \quad (1)$$

If $\Pr(G \in \mathfrak{G}^{(p)} \mid G \text{ is characterized by } S_j^{(p)})$ is not much different from $\Pr(G \in \mathfrak{G}^{(p)})$, i.e., whether or not G is characterized by $S_j^{(p)}$ makes very little difference, then $S_j^{(p)}$ should not be considered very interesting in determining if G should be classified into $\mathfrak{G}^{(p)}$. Otherwise, $S_j^{(p)}$ can be very interesting.

To objectively determine if the two probabilities are different, we make use of an adjusted residual, d_{jp} which is defined as

$$d_{jp} = \frac{z_{jp}}{\sqrt{\gamma_{jp}}} \quad (2)$$

where z_{jp} is defined as:

$$z_{jp} = \frac{\Pr(G \in \mathfrak{G}^{(p)} \mid G \text{ is characterized by } S_j^{(p)}) - n \Pr(G \in \mathfrak{G}^{(p)}) \Pr(G \text{ is characterized by } S_j^{(p)})}{\sqrt{n \Pr(G \in \mathfrak{G}^{(p)}) \Pr(G \text{ is characterized by } S_j^{(p)})}} \tag{3}$$

and γ_{jp} is the maximum likelihood estimate of the variance of z_{jp} and is given by

$$\gamma_{jp} = (1 - \Pr(G \in \mathfrak{G}^{(p)}))(1 - \Pr(G \text{ is characterized by } S_j^{(p)})) \tag{4}$$

Based on [12], if $|d_{jp}| \geq 1.96$, we can conclude that the difference between $\Pr(G \in \mathfrak{G}^{(p)} | G \text{ is characterized by } S_j^{(p)})$ is significantly different from $\Pr(G \in \mathfrak{G}^{(p)})$ and therefore the subgraph $S_j^{(p)}$ is interesting and useful for classification. If $d_{jp} \geq +1.96$, it implies that the presence of $S_j^{(p)}$ in a graph G provides evidence supporting G to be classified into $\mathfrak{G}^{(p)}$ otherwise if $d_{jp} \leq -1.96$, it implies that the presence of the frequent subgraph $S_j^{(p)}$ provides negative evidence against G to be classified into $\mathfrak{G}^{(i)}$. In either case, $S_j^{(p)}$, can be considered an *interesting* frequent subgraph.

With the use of the adjusted residual analysis, *MISPAG-FP* screens each set of frequent subgraphs, $\mathbf{S}^{(p)} = \{ S_1^{(p)}, \ldots, S_s^{(p)}, \ldots, S_{m_p}^{(p)} \}$, $p = 1, \ldots, P$, to retain only those who are interesting. The set of interesting frequent subgraph discovered for each of $\mathfrak{G}^{(1)}, \ldots, \mathfrak{G}^{(p)}, \ldots, \mathfrak{G}^{(P)}$ respectively is denoted as $\mathbf{S}'^{(p)} = \{ S_1'^{(p)}, \ldots, S_j'^{(p)}, \ldots, S_{m_p'}'^{(p)} \}$, $p = 1, \ldots, P$, and $m_p' < m_p$. The algorithm of interestingness measure of *MISPAG* is given in *Figure* 9.

Algorithm of interestingness measure of *MISPAG*

Input : Input graph $\mathfrak{G}$, a set of frequent subgraphs **S**, (a query graph G_q)

Output : A set of interesting frequent subgraphs **S**'

for each $\mathbf{S}^{(p)} \in \mathbf{S}$ **do**

for each $S_j^{(p)} \in \mathbf{S}^{(p)}$ **do**

if $S_j^{(p)} \notin \mathbf{S}^{(m)}$

for each $\mathfrak{G}^{(m)} \in \mathfrak{G}$ where $m \neq p$ **do**

for each $G \in \mathfrak{G}^{(m)}$ **do**

if G contains $S_j^{(p)}$

Increment the count of $S_j^{(m)}$ by 1

else

store count of $S_j^{(m)}$

for each $S_j^{(p)}$ **do**

Calculate the expected frequency of $S_j^{(p)}$

Calculate the standard score z_{jp} of $S_j^{(p)}$

Calculate the maximum likelihood estimate of variance γ_{jp} of $S_j^{(p)}$

Calculate the test statistic d_{jp} of $S_j^{(p)}$

if $|d_{jp}| \geq 1.96$

Add $S_j^{(p)}$ to **S**'

return S'

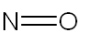
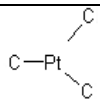
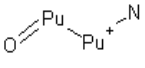
Figure 9. Algorithm of MISPAG

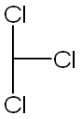
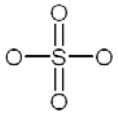
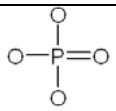
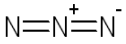
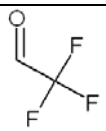
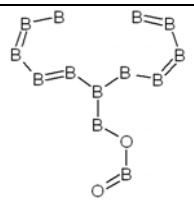
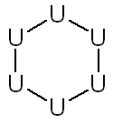
3.4 The Illustrative Example Continued

To illustrate how the interestingness measure of *MISPAG-FP* works, let us consider the example in Section 3.2 again. Given the frequent subgraphs discovered using *FSG* at a support threshold of 50%, *MISPAG-FP* obtains for each of the 15 frequent subgraphs their frequency of occurrences in each class. It then screens for all frequent subgraphs that are interesting using the interestingness score that is calculated by the test statistics given as Equation (2). The value of the test statistics for each frequent subgraph in each class are given also in *Table* 7.

As described in the last section, subgraphs with $|d_{jp}| < 1.96$ will be filtered out, and the remaining subgraphs will form a set of interesting subgraphs for graph classification. Since d_{41}, d_{51}, d_{62}, d_{72}, d_{83}, d_{93} are greater than 1.96, we conclude that, of all 15 frequent subgraphs discovered, only $S_4^{(1)}$ and $S_5^{(1)}$, $S_6^{(2)}$ and $S_7^{(2)}$, and $S_8^{(3)}$ and $S_9^{(3)}$ are interesting frequent subgraphs for each of *Class* 1, 2 and 3 respectively.

Table 7. Occurrence and Interestingness measure of frequent subgraphs (σ= 50%)

o_{jp} / d_{jp}		*Class* 1	*Class* 2	*Class* 3
$S_1^{(p)}$		8 0.14	8 -0.16	8 0.02
$S_2^{(p)}$		4 -0.94	7 0.39	7 0.54
$S_3^{(p)}$		6 0.69	6 0.44	3 -1.13

$S_4^{(p)}$		5 2.28	1 -1.16	1 -1.08
$S_5^{(p)}$		5 2.28	1 -1.16	1 -1.08
$S_6^{(p)}$		1 -1.03	5 2.08	1 -1.08
$S_7^{(p)}$		1 -1.03	5 2.08	1 -1.08
$S_8^{(p)}$		1 -1.03	1 -1.16	5 2.19
$S_9^{(p)}$		1 -1.03	1 -1.16	5 2.19
$S_{10}^{(p)}$		1 -1.54	4 0.36	5 1.16
$S_{11}^{(p)}$		5 0.98	2 -1.19	4 0.23
$S_{12}^{(p)}$		5 1.25	4 0.36	1 -1.60
$S_{13}^{(p)}$		4 -0.10	4 -0.32	5 0.42
$S_{14}^{(p)}$		4 -0.10	5 0.29	4 -0.19
$S_{15}^{(p)}$		5 0.51	4 -0.32	4 -0.19

3.5 Experiments and Results

To evaluate the effectiveness of our approach, it is tested using both artificial and real data. We compared its performance with that of two graph classification algorithms based on *FSG* and *gSpan*. We used the executable files of these algorithms available from [118] and [119] respectively. The classification results were obtained using 10-fold cross validations with an implementation of SVM available at [120].

The performance of a classifier is usually evaluated by the use of average classification accuracy and the results are typically presented in a confusion matrix (*Table* 8) which has four entries: the number of true positive cases (*TP*), true negative cases (*TN*), false positive cases (*FP*) and false negative cases (*FN*) and the average accuracy is calculated as follows: [121]

$$\text{Average Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{5}$$

Table 8. Confusion Matrix

		Predicted	
Actual		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

While evaluation based on the use of the classification accuracy measure may be popular, it may not always be very appropriate for classification problems involving imbalanced class distributions. When *TN* is much greater than *TP*, (*FP* + *TN*) is also much greater than (*TP* + *FN*). In such case, the successfully predicted cases in the minority positive class will play a role that can be too insignificant when the average accuracy rate is determined and the minority cases will be treated as noise even if they are supposed to be important. In order to overcome this problem, the true positive and false positive rates need to be monitored separately using Equation (6) and (7) when test data are being classified.

$$\text{True positive rate} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{False positive rate} = \frac{FP}{FP + TN} \tag{7}$$

These rates measure the performance of a classifier for each class and the objective is to keep the true positive rate as high as possible and the false positive rate as low as possible. Sometimes, the true positive rate is called recall or sensitivity, and the false positive rate is called false alarm rate. In order to transform this multi-objective problem into a single-objective equivalent, the ROC analysis [122] has been proposed and is becoming more and more popular when the training data size for different classes of data are very different. With the ROC analysis, the true positive rate is plotted along the y-axis against the false positive rate along the x-axis

to form a ROC curve, and the objective is to maximize the value of AUC which stands for the Area Under the ROC Curve. The value of AUC is always between 0.0 and 1.0. An area of 1 represents a perfect classification, whereas an area of 0.5 represents a worthless classification that is equivalent to a random guess in a two-class classification problem. The AUC is an important statistical property that is equivalent to the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance. In this paper, as the data sets that we use differ significantly in class sizes, we will use the AUC to evaluate the performance of different classifiers on different datasets.

3.5.1 Datasets

The first dataset is a set of binary-class artificial data that are generated with GraphGen [31]. The artificial datasets are generated with a set of parameters: 1) the total number of transactions (-ngraphs), 2) the average size of each graph (-size), 3) the number of unique node labels (-nnodel), 4) the number of unique edge labels (-nedgel), 5) the average density of each graph (-density), 6) the number of unique edges in the whole dataset (-nedges), and 7) the average edge ratio of each graph (-edger). The parameter 1, 4, 5, 6 and 7 are fixed to respectively 5000, 10, 0.3, 100 and 0.2 and we vary the remaining parameters to generate four datasets as given in *Table* 9 with properties below.

Table 9. Artificial dataset with different parameters

Dataset	-size	- nnodel
D1	10	5
D2		10
D3	30	5
D4		10

The second dataset is collected from Predictive Toxicology Challenge (PTC) [123] that contains the carcinogenicity of 417 chemical compounds on four types of rodents: male rats (MR), female rats (FR), male mice (MM) and female mice (FM). Each of these data sets can be considered as consisting of two classes of data [124]: those with positive evidence of cancerous growth and those with negative evidence.

The third dataset is collected from the Estrogen Receptor Binding (NCTR ER) database in the Distributed Structure-Searchable Toxicity (DSSTox) Public Database Network of the National Center for Toxicological Research [125]. The database covers most known estrogenic classes and it is a structurally diverse set of estrogens. The NCTR ER database consists of 224 chemical compounds with each classified as active or inactive with respect to the attribute "ActivityOutcome_NCTRER". A compound is active if the measure of activity of the compound is active strong, medium or weak. It is inactive if there is no activity for that compound. The properties of the datasets we used in our experiments are listed in *Table* 10.

Table 10. Properties of the experimental datasets

Dataset	Total no. of sample	Percentage of positive samples
D1, D2, D3, D4	5000	40.0%
MR	322	37.9%
FR	334	31.1%
MM	308	35.1%
FM	331	38.1%
NCTR ER	224	58.5%

3.5.2 Performance Analysis

For performance comparison, we tested all datasets using first the two algorithms of *FSG*, *gSpan* and then compare their performance when *MISPAG-FP* is used. *Table* 11 shows the performance of each algorithm on the different datasets. For easier comparisons, we use a single misclassification cost value of 3.0 and as suggested in [126] for the SVM classifier.

For our experiments, as a high threshold may result in too little and a low threshold may result in too many of the frequently-occurring subgraphs being discovered, and as the support threshold is proportional to the runtime and memory consumption [127], we tried different support thresholds ranging from 90% to 2% and decided to settle at 3% for the artificial dataset, 5% for the PTC dataset, and 10% for the NCTR ER dataset for both the experiments with *FSG* and *gSpan*. These

settings allow us to obtain a good size of subgraphs (i.e. $50 \leq n \leq 500$) for the identification of the interesting ones.

Given these settings of the support thresholds, the average AUC for each algorithm is determined and shown in the *Table*. From these results we can see that the classification performance (average AUC) of *FSG* and *gSpan* are similar. The average AUC for them are 0.673 and 0.691 respectively. After applying *MISPAG-FP* to these frequent subgraph discovery algorithms, their average AUC improved by 14.44% and 14.05% respectively.

Table 11. Classification performance for FSG, gSpan and MISPAG-FP

Dataset	AUC		Percentage Improvement	AUC		Percentage Improvement
	FSG	*MISPAG-FP* with *FSG*		*gSpan*	*MISPAG-FP* with *gSpan*	
D1	0.778	0.897	15.30%	0.780	0.906	16.15%
D2	0.823	0.904	9.84%	0.854	0.912	6.79%
D3	0.623	0.874	40.29%	0.621	0.877	41.22%
D4	0.778	0.897	15.30%	0.786	0.908	15.52%
MR	0.603	0.642	6.47%	0.605	0.644	6.45%
FR	0.505	0.587	16.24%	0.517	0.592	14.51%
MM	0.569	0.603	5.98%	0.572	0.594	3.85%
FM	0.634	0.697	9.94%	0.635	0.699	10.08%
NCTR ER	0.835	0.924	10.66%	0.852	0.953	11.85%
Average	**0.683**	**0.781**	**14.44%**	**0.691**	**0.787**	**14.05%**

These results show that the performance of *FSG* and *gSpan* can be improved with the two-phase approach that *MISPAG-FP* adopts. The subgraphs discovered by many graph mining algorithms may appear frequently in a class but they may not uniquely represent a class. Subgraphs that may not appear very frequently can play an important role in discriminating one class from another. With

MISPAG-FP, the relative frequency of each subgraph is considered and how useful they are for classification are determined with a measure. The measure is then used when a graph is classified. This makes *MISPAG-FP* more effective a graph classification algorithm.

The datasets D1 to D4 are the artificial dataset with varied size of graph samples and number of unique node labels. When the number of unique node labels is increased from 5 to 10, we can see that the classification performance is higher for D2 with more unique node labels than D1 with less unique node labels, the case is the same for D3 and D4. The reason is that the combination of the discovered frequent subgraphs will be less if the number of unique node labels is small. For example, if there are only two node labels v_1 and v_2 in the dataset, we have only three combinations (v_1–v_1, v_1–v_2, v_2–v_2) for a graph with two vertices and one edge; if there are five node labels v_i where i = 1 to 5 in the dataset, we can have 15 combinations. In the case with less unique node labels, many frequent subgraphs will be the same for both positive and negative class. These frequent subgraphs are uninteresting and not useful in discriminating the graph sample into different classes. With *MISPAG-FP*, we can filter those uninteresting frequent subgraphs to increase the classification performance. Hence, we can observe from the results that the average AUC of D1 is lower than that of D2, and the AUC is increased more significant in D1 than D2 after applying *MISPAG-FP*. When the size of graph samples is increased from 10 to 30, we can see that the classification performance is lower for D4 with larger graph size than D2 with smaller graph size, the case is the same for D1 and D3. The reason for this is that a large graph will contain more noise

than a small graph as the interesting subgraph(s) usually contribute a small part in a graph. From the results, we can see that the average AUC of D4 is lower than D2, and *MISPAG-FP* helps to remove those noisy frequent subgraphs and increase the AUC more significantly in D4 than D2 as the graph size in D4 is larger than that of D2.

The PTC dataset contains four datasets: MR, FR, MM and FM. The average AUC of FM is the highest and that of FR is the lowest. This may be due to the percentage of the positive samples of FM (38.1%) being higher than that of FR (31.1%). The overall AUC for the PTC dataset is 0.58 when applying *FSG* and *gSpan*, and this value has increased to 0.63 with *MISPAG-FP*. The overall AUC is still relatively low even when *MISPAG-FP* is used and this may be due to some structural features in the test set not being present in the training set. This is the main reason that the classification performance is quite low. This phenomenon is also mentioned in the evaluation report of [123].

The NCTR ER dataset has the highest AUC throughout the experimental datasets. The average AUC for *FSG* and *gSpan* is 0.844 and this is increased to 0.939 with *MISPAG-FP*. This means that the ER compounds contains distinguishing structures for active and inactive classes. With the discovered interesting frequent subgraphs, we can use them to characterize a class of estrogen as well as discriminate a sample from another in a different class. From the percentage of improvement in AUC, we can observe that the noisy and uninteresting frequent

subgraphs are effectively screened by *MISPAG-FP* and the AUC is maximized when it is used with *FSG* and *gSpan*.

3.6 Summary

In this chapter, we introduced a new graph mining approach called *MISPAG-FP* to discover interesting frequent subgraphs from graph databases. We used *MISPAG-FP* by combining it with other frequent subgraph discovery algorithms on both artificial and real datasets to test the effectiveness and performance. The experimental results show that *MISPAG-FP* can work very well with large and complex datasets and can improve the classification performance of the existing graph mining algorithms.

The frequent subgraphs of real molecular networks usually contain many common vertices (e.g. carbon C and oxygen O) and edges (e.g. single hydrogen bond). For this reason, both positive and negative samples may contain the same set of frequent subgraphs. The frequent subgraphs discovered by existing graph mining algorithms may therefore not be very useful for molecular classification. *MISPAG-FP* is able to achieve a higher accuracy as it aims to discover the "interesting" subgraphs which may not necessarily be the most frequently occurring ones. *MISPAG-FP* can better handle the problem of having too many frequent subgraphs when support-thresholds are lowered. Like other graph mining algorithm, the size and number of graphs that *MISPAG-FP* can handle can be very large and they are limited mainly by computing hardware.

Chapter 4

Discovering Interesting Structural Patterns with Applications to Motif Discovery in PPI Networks

The problem with *MISPAG-FP* is that it still relies on a frequent subgraph mining algorithm to discover frequently occurring subgraphs first. If an interesting subgraph does not occur frequently enough, if would not be discovered in the first place. As a result, we need an algorithm that can discover interesting subgraphs even if they do not occur frequently. In this section, we describe such an algorithm and explain how it can be used to discover motifs in protein-protein interaction networks.

Proteins interact with each other in a cell, and these interactions are represented as a protein-protein interaction (PPI) network. Protein-protein interactions are of great interest because they are involved in virtually all cellular processes and are amongst the most ubiquitous types of interactions. Various high-throughput techniques including yeast two-hybrid system [92], mass spectrometry [93], protein microarray [94], and synthetic lethality screen [95] are used to generate large-scale PPI data. More public repositories [14, 15, 16, 17] are made available online to get access to these large-scale data and facilitate the analysis of PPI.

Interacting proteins are likely to collaborate to form a network motif. Network motifs are connected subgraphs that act as the basic building blocks of complex networks as suggested in [24]. Hartwell et al. [25] suggested that

understanding the relatedness of protein function in PPI networks is useful because knowledge about protein can inform the function of the other related proteins, and the modular structure can further facilitate the understanding of such biological processes as cellular organization, transcription regulation and phenotypic evolution, etc. It is believed that such network motifs are useful to describe the structure and functionality of a PPI network. As a PPI network can be naturally represented as an undirected graph such that each vertex represents a protein and each edge represents the interaction between two vertices, subgraph mining approach is suitable for the task of motif discovery.

Recently, there has been much effort to analyze the protein-protein interaction (PPI) network in *S. cerevisiae* for network motif discovery [128, 129], [130, 131], protein function prediction [132, 133], protein complexes identification [134, 135], etc. As a cell of *S. cerevisiae* contains thousands of interacting proteins whose functions are dependent on many other proteins that it interacts with, its structure can be very complex. If hidden regularities can be discovered in them, how a protein interacts with the other proteins to perform a certain function or how related proteins interacts to form signaling proteins or drug targets [136, 137, 138], etc., can be more easily determined and understood.

4.1 Discovering Interesting Subgraphs

Like other PPI networks, the PPI network of *S. cerevisiae* can be represented as a graph with vertices representing proteins and edges representing interactions

between proteins. Given such a representation, a number of algorithms [139, 140, 141, 142, 143, 144] have been proposed to detect for conserved patterns in the PPI network of *S. cerevisiae*. These algorithms are either based on the use of network alignment techniques [139, 140, 141] or the search for frequent subgraphs [142, 143, 144].

Algorithms based on network alignment aim at identifying conserved network regions in the graphs corresponding to the PPI network of *S. cerevisiae* and that of another species. Given the two network graphs, an alignment graph is first constructed by finding pairs of corresponding homologous proteins in each of them and they are then represented as vertices. The edges connecting the vertices are then used to represent conserved interactions. The network alignment approach does not scale up very well when handling large PPI networks, and the number of vertices in an alignment graph can grow exponentially in size with the number of aligned networks and this can make network alignment infeasible [139, 141]. To tackle this problem, there have been some attempts to discover conserved patterns by discovering frequent subgraphs across the network graphs using graph mining algorithms.

The PPI network of *S. cerevisiae* is composed of interacting modules with different functions [r8, r9]. These functional modules are expected to repeat in the network. By discovering frequently occurring subgraphs, conserved patterns in the PPI network can be identified. This is why many graph mining algorithms that can be used to discover frequent subgraphs in molecular graphs are used to discover

conserved patterns in PPI networks. For example, Koyuturk et al. developed an algorithm called MULE [142] to mine frequently occurring edgeset patterns using a depth-first search (DFS) approach to locate all connected subgraphs in a graph database. MULE has been used with eukaryotic PPI networks, including that of *S. cerevisiae*, to relate the functions of proteins in different organisms. Chen et al. developed gApprox [143] to mine frequently occurring approximate patterns with tolerable variations as measured by a minimum support and a maximal number of disjoint occurrences. The degree of approximation of a pattern is defined in terms of some vertex and edge penalties measures. These measures are functions of the dissimilarity and tightness association between proteins. gApprox has been used with eukaryotic PPI network data to discover pairs of functionally similar proteins in similar locations in the network. Other than gApprox, Turanalp et al. developed PPISpan [145] to mine for frequently occurring functional interaction by combining the process of candidate pattern generation and pruning to speed up the graph mining process. Borgwardt et al. [146] compares PPI networks using a frequent subgraph mining approach that excludes subgraphs that appear only in a negligibly small fraction in a data set. This approach has been used with PPI networks of *S. cerevisiae* to identify frequently occurring subgraphs in the functional space.

While graph mining algorithms can be used to discover patterns in PPI networks, they have mainly been used to discover subgraphs in network graphs that occur frequently enough. However, subgraphs that occur frequently may not necessarily be interesting and be biologically meaningful. Discovering frequently-occurring subgraphs, therefore, may not reveal all interesting patterns in a PPI

network. For example, protein complexes, which do not usually repeat themselves in a PPI network, will not be discovered with many graph mining algorithms. What is needed is therefore a graph mining algorithm that can discover meaningful structural patterns rather than just the most frequently occurring ones.

4.2 Discovering Interesting Structural Motifs

We propose an algorithm called *MISPAG-DM* to discover sets of interesting motifs from the PPI networks. We perform the tasks by first constructing a network graph from this PPI network. This graph G can be denoted as G(**V**, **E**) where **V** and **E** are the sets of vertices and edges respectively. Each vertex in **V** represents a protein in the PPI network and each edge in **E** represents the existence of an interaction relation between the two proteins that it connects with. To perform its tasks, *MISPAG-DM* examines each vertex in turn to determine if its interactions with each of its neighboring vertices are interesting. If so, the protein pairs forms part of a larger interesting subgraph. Otherwise, they are not considered for further processing. To determine if a vertex and any of its neighboring vertices is interesting, an objective interestingness measure that is introduced in Chapter 3 is used. The flowchart of *MISPAG-DM* is given in *Figure* 10.

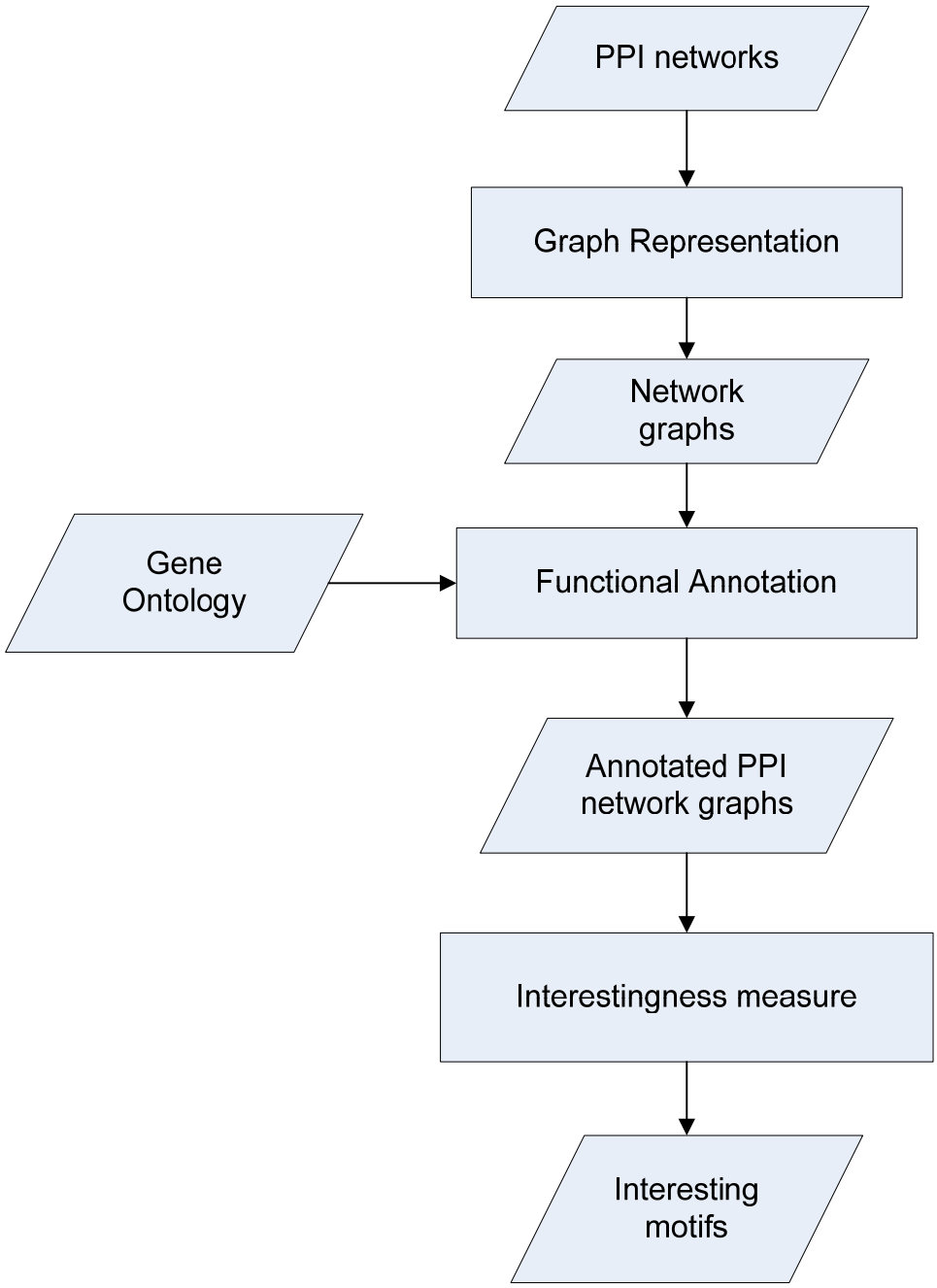


Figure 10 Flowchart of MISPAG-DM to discover interesting motifs from PPI networks

4.2.1 Representing PPI networks in Graphs

The PPI network of *S. cerevisiae,* which is used for this work, is obtained from the Database of Interacting Proteins (DIP) [147]. It is made up of 1361 proteins and 3222 experimentally confirmed interactions between proteins. Graph representation is widely used to model the biological networks as well as other biological domains, and it is suitable for the description of complex structures in such biological networks as PPI networks.

A PPI network is defined as an undirected graph, an ordered pair $G = (\mathbf{V}, \mathbf{E})$, where **V** is a set of vertices representing proteins and **E** is a set of binary edges representing interactions between proteins. Each edge $e \in \mathbf{E}=\{e_1, ..., e_{m_e}\}$ contains two ends v and $v_i \in \mathbf{V}=\{v_1, ..., v_{m_v}\}$, and v and v_i are incident with e and neighbors to each other. The vertices incident to an edge are called its end-vertices, and the degree k of a vertex v is the number of edges that have v as an end-vertex. Some proteins may interact with itself and if this is the case, it will be denoted by an edge that has the same end-vertices and we will then have a loop. *Figure* 11 shows an example PPI network with five proteins and eight interactions between them. It can be modeled as a graph G with vertex set **V** = {P32492, P19524, P08964, P53141, P36126} and edge set **E** {{P32492, P19524}, {P32492, P08964}, {P19524, P08964}, {P19524, P53141}, {P08964, P53141}, {P53141, P36126}}.

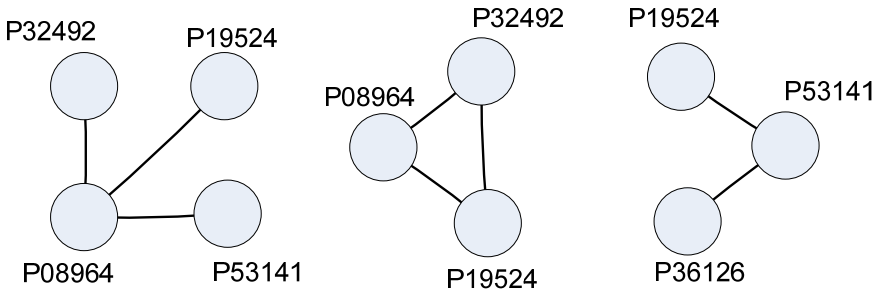


Figure 11 Graph representation of an example PPI network with five proteins and eight interactions

In a network graph, not all vertices are connected with each other and some of them may form a disjoint set (e.g. the vertex sets: {P32492, P19524, P08964, P53141}, {P32492, P19524, P08964} and {P19524, P53141, P36126}), which is a special kind of subnetwork. A subnetwork S = ($\mathbf{V}_S$, $\mathbf{E}_S$) of the network graph G = (**V**, **E**) is a subgraph where $\mathbf{V}_S \in \mathbf{V}$ and $\mathbf{E}_S \in \mathbf{E}$. A subnetwork is not necessarily a disjoint set of a graph, whereas it can be a connected subgraph from any part of a network. *Figure* 12 shows a subnetwork with three interacting proteins occurring twice in G.

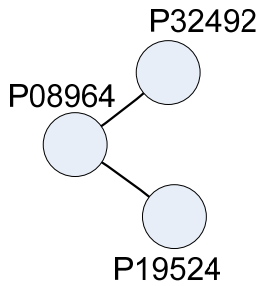


Figure 12 A subnetwork S of the network graph G with two occurrences

4.2.2 Functional annotation with gene ontology

The raw data of biological network is a labeled graph that treats each biomolecule as a unique entity and it does not contain other useful biological information. Since the functions of some biomolecules like proteins and genes are already annotated in databases such as Gene Ontology (GO) [8], we can model each network as functional annotated template for further analysis. Functional annotation of biological networks helps to define, understand and compare essential cellular actions in different organisms that involve similar functional units. It also helps to determine the biological functional roles of the other un-annotated units. For the purpose of pattern discovery, instead of the name or ID of each protein, the molecular function it performs is considered in the graph mining process. We label the proteins in the PPI network graph with the molecular function of the GO database, so that each protein will be assigned with its corresponding functional GO term. For example, the vertices in the PPI network graph of *S. cerevisiae* in *Figure* 13, which are used to represent the proteins of Swiss-Prot:P02294, Swiss-Prot:P22276, Swiss-Prot:P32349, and Swiss-Prot:P04051 are labeled with their functions "*DNA binding*", "*protein binding*", "*phosphatidylinositol-4,5-bisphosphate binding*", and "*RNA binding*" respectively.

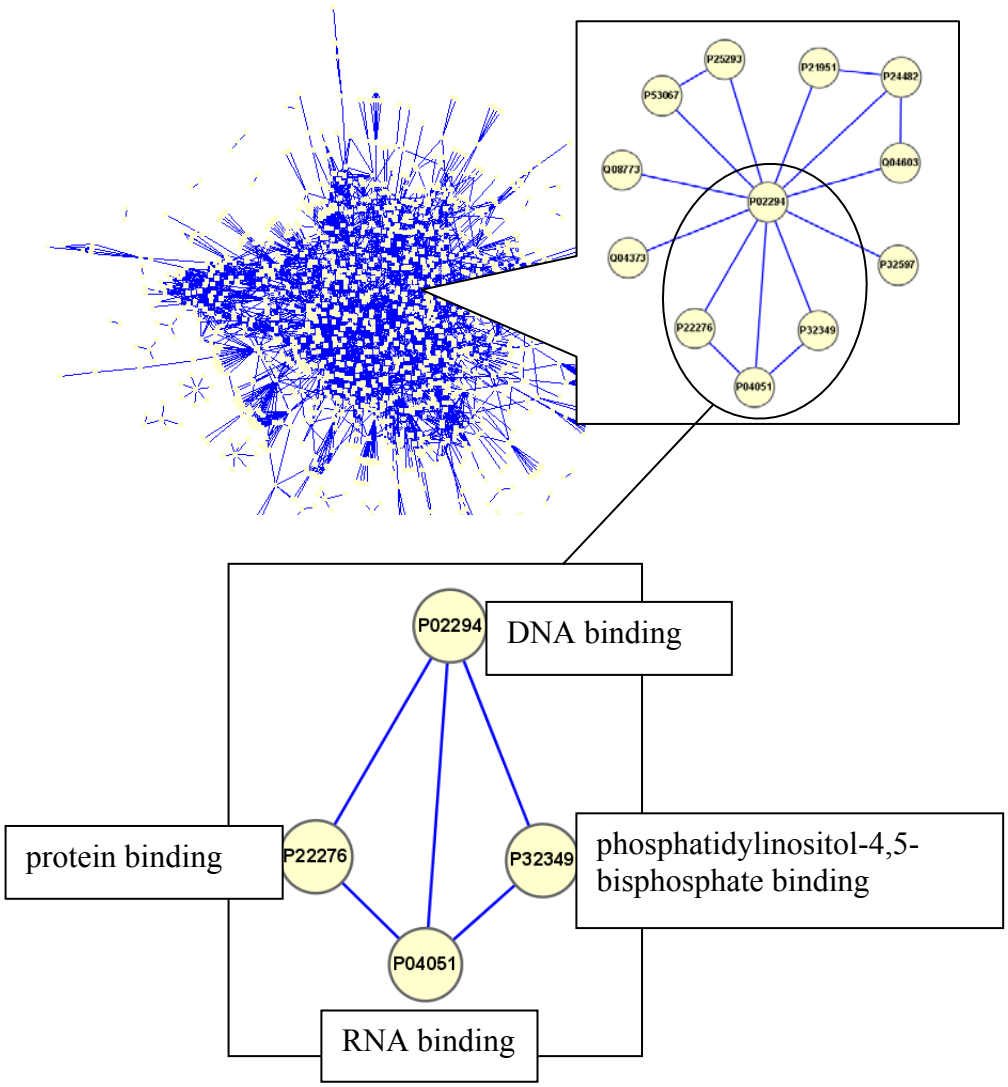


Figure 13 A labelled molecular graph representation of the PPI network of S. cerevisiae

Once the network graph G, labeled with the functions of its constituent proteins, is constructed, we proceed to discover interesting patterns in it. To match these functions to the biomolecules, a matching index of identifier and molecular functions is constructed to facilitate the function matching. However, a biological unit may represent in different ID in different databases [14, 15, 68, 69, 70], for

example, P53141 in IntAct, 5576N in DIP, YGL106W in Ensembl and KEGG. For those biological network databases that are not identified by the UniProtKB/Swiss-Prot ID, mappings are developed between the databases in order to locate the corresponding functions in GO.

4.2.3 The motif-discovery algorithm in details

To discover interesting structural motifs from one or more functionally annotated PPI network graphs, each candidate subgraph is verified with the use of an interestingness measure to determine if it appears more frequently than expected. If not, the subgraph is uninteresting and will be screened away. The algorithm of applying *MISPAG-DM* on PPI networks can be summarized in *Figure* 14.

Firstly, a frequency vector matrix and adjusted residual matrix are initialized for calculating the interestingness measure. By considering a confidence level of 95 percent, the interestingness threshold μ is set to 1.96 for validation. A set of distinct one-edge subgraphs will be extracted from each functionally annotated PPI graph so that no duplicate subgraph will co-exist in the one-edge subgraph list $\mathbf{S}^1$. The frequency of the candidate subgraph g in $\mathbf{S}^{k-1}$ is counted with the use of a depth-first search from the network graph G and the frequency value is stored in the frequency vector matrix for the calculation of adjusted residual value.

Let us consider a vertex $v \in \mathbf{V}$. Assuming that the protein that v represents interacts with n other proteins, then v can be considered as connected to n other vertices, $v_1, v_2, ..., v_n$. To determine if the interaction between v and $v_i \in \{v_1, v_2, ..., v_n\}$

is interesting, we consider how frequently v occurs in **G** given that it is connected with v_i and compare it with how frequently it occurs in **G.** If the difference is significant, it means that the pattern occurs more or less frequently than expected and the interaction between v and v_i can therefore be considered interesting and it can form part of an interesting subgraph. In other words, we are interested in determining if the difference between the following conditional probability:

Pr(the vertex on one side of an edge is v | the vertex on the other side is v_i)

$$= \Pr(v \mid v_i) = \frac{\text{Total number of edges in G connecting } v \text{ and } v_i}{\text{Total number of edges in G that connect } v_i \text{ to other vertices}} \quad (8)$$

and the following apriori probability

Pr(the vertex on one side of an edge is v)

$$= \Pr(v) = \frac{\text{Total number of edges in G that connects } v \text{ to the other vertices}}{\text{Total number of edges in G}} \quad (9)$$

is significantly different.

If the difference is significant, the interaction between the proteins corresponding to v and v_i is considered interesting. In other words, if the protein corresponding to v is found, the protein corresponding to v_i is more likely than the others to be found interacting with it.

To allow interestingness to be compared, an interestingness measure defined in terms of the two probabilities in Equation (8) and (9) are used here [12, 13]. This interestingness score is called *adjusted residual* [12] and is defined as follows as described in the last chapter:

$$d_{vv_i} = \gamma[\Pr(v \mid v_i) - \Pr(v)] \tag{10}$$

where $\gamma = \dfrac{1}{\Pr(v)\sqrt{\Pr(v)(1-\Pr(v))\Pr(v_i)(1-\Pr(v_i))}}$ and

d_{vv_i} has a standard normal distribution.

If the value of d_{vv_i} is large, we can conclude that the connection between the vertices v and v_i, and therefore the interaction between their corresponding proteins, is interesting. As the magnitude of d_{vv_i} can be considered as reflecting the strength. After obtaining all adjusted residual value of all candidate graphs, they are validated with the interestingness threshold. The candidate subgraphs with adjusted residual value d_{vv_i} greater than μ will form a set of interesting candidate subgraphs. These subgraphs will be expanded to form the next level candidate subgraphs until no qualified candidate subgraphs can be further discovered and no candidate subgraph can be generated. If no individual subgraph has a qualified adjusted residual value, the next level of candidate subgraph may jointly have higher value. The final collection of qualified candidate subgraphs represents a set of interesting subgraphs $\mathbf{S} = \{\mathbf{S}^{(1)}, \ldots \mathbf{S}^{(p)}, \ldots, \mathbf{S}^{(P)}\}$ for each specific class of network.

Input : A set of functional annotated PPI graphs $\mathfrak{G}$

Output : A set of interesting subgraphs **S**

V = frequency vector matrix

R = adjusted residual matrix

μ = 1.96 (95% confidence level)

for each $\mathfrak{G}^{(p)} \in \mathfrak{G}$ **do**

$\mathbf{S}^{1\,(p)}$ = all distinct 1-edge subgraphs in $\mathfrak{G}^{(p)}$

$\mathbf{S}^1 = \mathbf{S}^1 \cup \mathbf{S}^{1\,(p)}$

$k = 2$

while $\mathbf{S}^{k-1}$ is not null **do**

for each candidate g_j in $\mathbf{S}^{k-1}$ **do**

Initialize the count of g_j

for each G_i **do**

Count the frequency of g_j in G_i with DFS

Store the frequency count v_{jp}

D = all adjusted residual value

if $d_{jp} \in D > \mu$

Insert g_j into $\mathbf{S}^{k(p)}$

$\mathbf{S}^k = \mathbf{S}^k \cup$ all potential candidates of g_j with one edge growth in $\mathfrak{G}^{(p)}$

$\mathbf{S}^{(p)} = \mathbf{S}^{(p)} \cup \mathbf{S}^{k\,(p)}$

Increment k by 1

return S = $\{\mathbf{S}^{(1)}, \ldots \mathbf{S}^{(p)}, \ldots, \mathbf{S}^{(P)}\}$

Figure 14 Algorithm of MISPAG-DM

With the use of the interestingness measure as defined by Equation (10), the uninteresting edges are filtered and only those connecting vertices that are interesting

can be retained in the network graph. Once the interesting edges are identified, what are remaining forms different interesting subgraphs that can correspond to significant network motif or protein complexes, etc.

4.3 Experiments and Results

To discover interesting patterns in the PPI network of *S. cerevisiae*, we choose the high-throughput (genome scale) data made available in the Database of Interacting Proteins (DIP) [15] for our experiments. The DIP contains detailed information on 1361 proteins in *S. cerevisiae* and 3222 experimentally confirmed molecular interactions between them. Based on the proteins and their interaction relationship, a network graph is constructed as described above. The graph is then labelled with the molecular functions performed by the proteins in the PPI network. In the labeling process, it is found that about 7% of the proteins in the PPI network of *S. cerevisiae* are not functionally annotated and the functions of these proteins are marked as unknown.

Given the labeled network graph, *MISPAG-DM* is used to discover interesting motifs in the PPI network of *S. cerevisiae*. An example of the patterns that are discovered is given in *Figure* 15. The pattern discovered is represented in a subgraph which happens to be exactly the same as a known protein complex, *exosome complex* (MIPS:440.12.10), in *S. cerevisiae* listed in MIPS [14]. As shown in the figure, out of the seven proteins in the subgraph, six of them perform the same function of "*RNA binding*", and only protein (Swiss-Prot:Q08162) perform a

different function "*holo-[acyl-carrier-protein] synthase activity*". The interesting subgraph discovered shows that *MISPAG-DM* can discover protein complexes consisting of proteins that mainly interact with other proteins that perform the same functions.

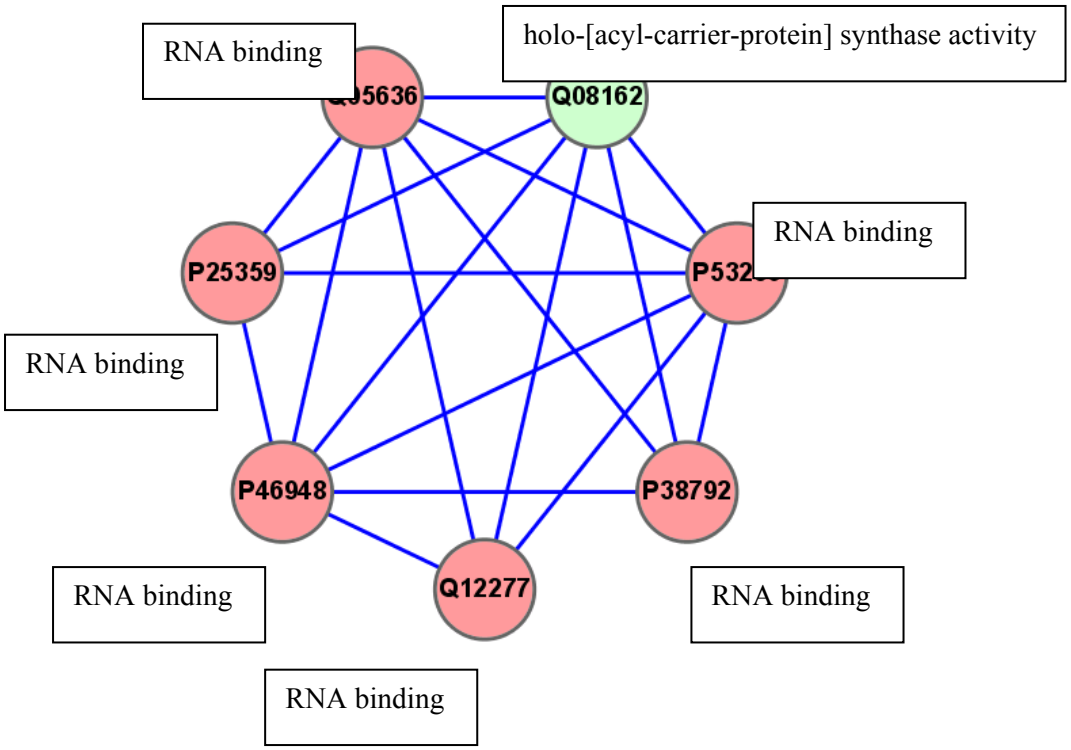


Figure 15 An interesting structural pattern that is made up of proteins that perform the same molecular function

Other than discovering interesting structural patterns that are made up of proteins that perform mainly the same functions, *MISPAG-DM* can also discover patterns that are made up of proteins that perform different functions. For example, *Figure* 16 shows an interesting subgraph of 10 vertices that are discovered by *MISPAG-DM*. It matches exactly with another known signaling protein complex

(MIPS:550.1.166) in *S. cerevisiae*. Unlike the pattern shown in *Figure* 2, these ten proteins perform totally different functions. This confirms that even if the proteins perform different functions, they can also interact with each other to form a complex.

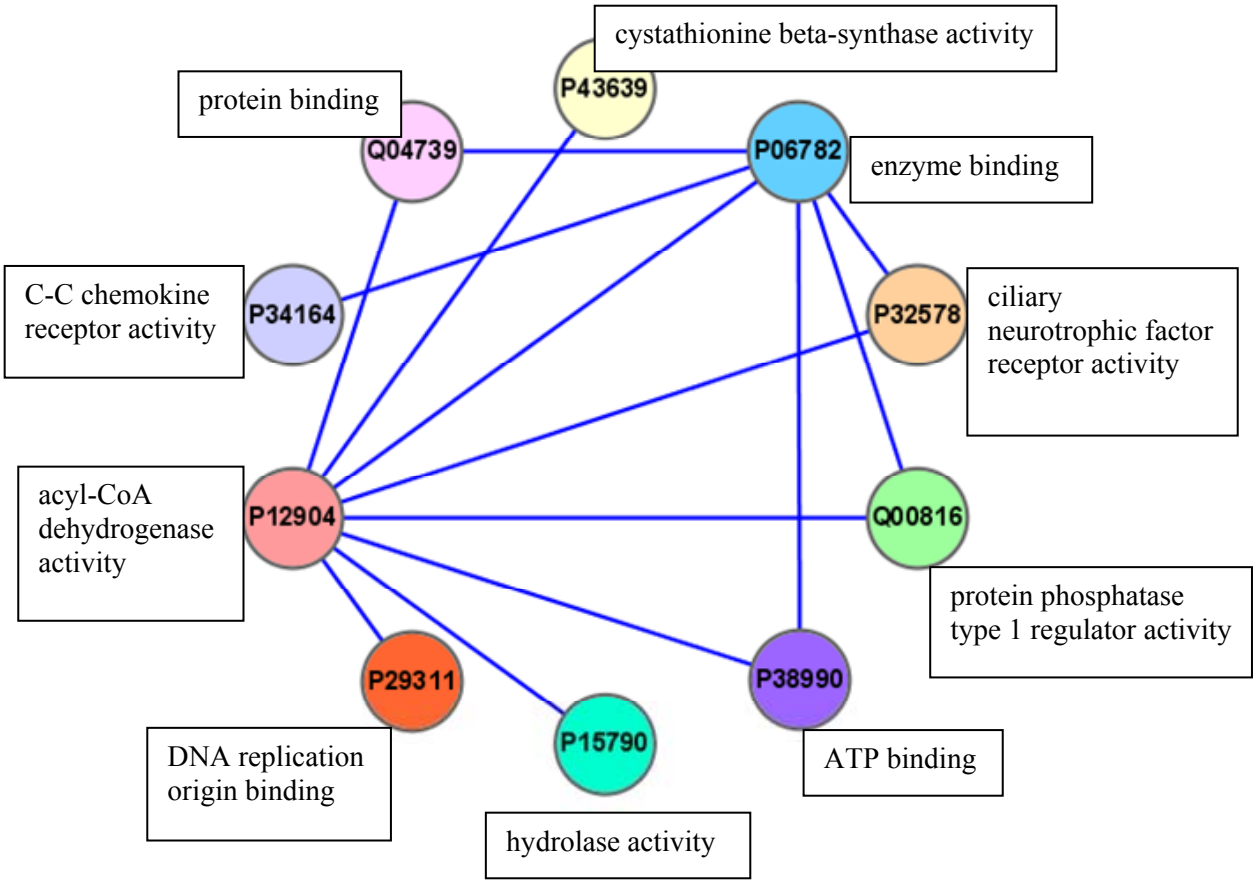


Figure 16 An interesting structural pattern that is made up of proteins that perform different molecular function

Many graph mining algorithms can be used to discover interesting patterns in the PPI network of *S. cerevisiae* that occur frequently enough. *MISPAG-DM* can also do so. In *Figure* 17, we show examples of frequently-occurring subgraphs discovered by *MISPAG-DM*. The frequent subgraphs discovered with graph mining

algorithms, such as *gSpan* [45], are usually small in size. This is because large subgraphs do not usually occur frequently enough. For example, in the case of the PPI network of *S. cerevisiae*, graph mining algorithms can only discover frequent patterns of up to three vertices. Larger interesting subgraphs such as those corresponding to many known protein complexes do not usually appear frequently enough for many graph mining algorithms to discover.

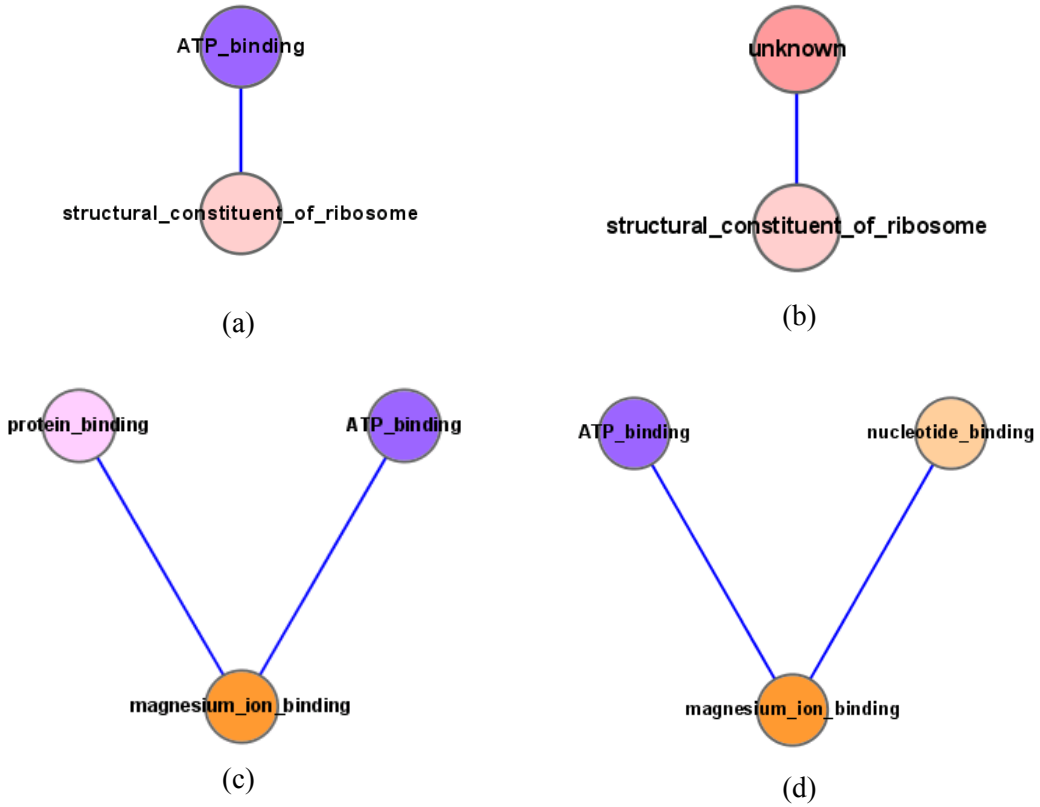


Figure 17 Example of Frequent Subgraphs discovered by graph mining algorithms in the PPI network of S. cerevisiae

For *MISPAG-DM*, however, it can not only discover patterns that occur frequently but can also discover patterns that are biologically meaningful but do not occur frequently enough. An example of it is shown in *Figure* 18. The subgraph

discovered by *MISPAG-DM* matches exactly with that of a known protein complex (MIPS:410.10] in *S. cerevisiae*. This subgraph cannot be discovered with graph mining algorithms as it only occurs once in the PPI network of *S. cerevisiae*. This complex is called "*post-replication complex*" which is made up of six proteins (Swiss-Prot: P32833, P38826, P54784, P54790, P50874, and P54791). Among these proteins, they perform four different kinds of molecular functions.

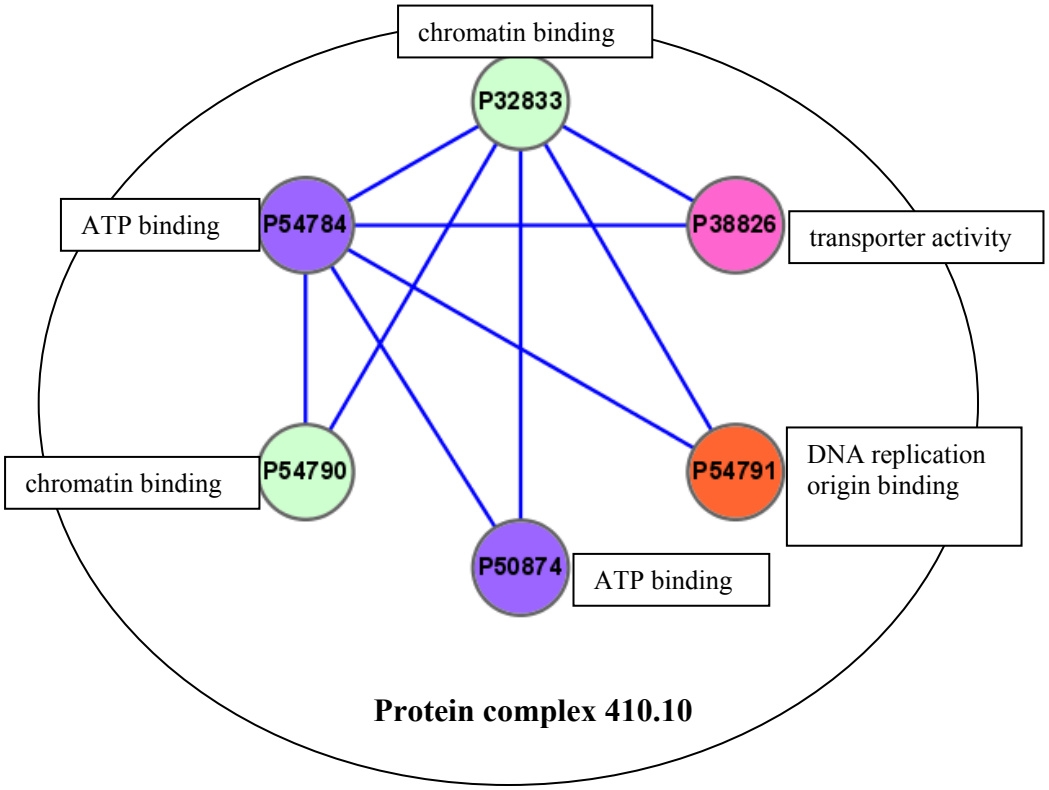


Figure 18 An interesting structural pattern discovered by MIPIG that match exactly with a known protein complex

In *Figure* 19, we show another interesting subgraph discovered by *MISPAG-DM*. This subgraph has seven vertices. As opposed to the case shown in *Figure* 18, the subgraph discovered does not match exactly with that of known

protein complexes in *S. cerevisiae*. However, it matches mostly with another known protein complex (MIPS:260.90) called "*Arp2p/Arp3p complex*" which is made up of six proteins (Swiss-Prot: Q05933, P53731, P32381, P40518, P47117, and P33204), and the interesting structural pattern that we discovered contains seven proteins in which six of them match exactly with this complex. We found that the most abundant function in this complex is "*actin binding*" which is the essential for the actin filament polymerization in the *Arp2p/Arp3p complex*. Again, the subgraph does not occur any more than once and cannot be discovered with graph mining algorithm.

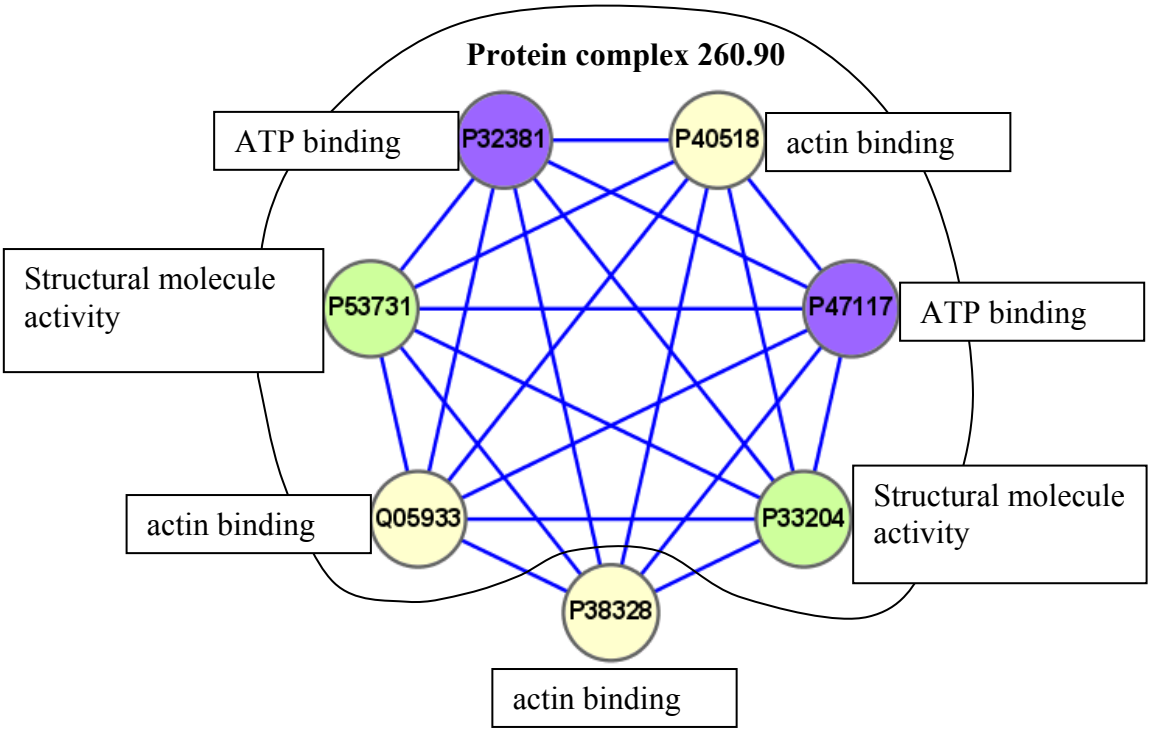


Figure 19 An interesting structural pattern discovered by MIPIG that matches partially with a known protein complex

4.4 Summary

Proteins interact with the other proteins in a PPI network. If interesting structural patterns can be discovered in such network, one may be able to better understand what proteins interact and how they interact with each other to perform a variety of cellular processes such as metabolic cycles, DNA transcription and replication, etc. [148, 149]. Traditional graph mining algorithms discover structural patterns in PPI networks by discovering frequently occurring subgraph in their network graphs. They do not consider interestingness of the protein interactions and cannot discover interesting subgraphs that do not occur frequently enough. To discover structural patterns that do not occur frequently enough, *MISPAG-DM* is proposed here. *MISPAG-DM* is able to discover interesting protein interactions and based on them, it can discover interesting structural patterns in PPI networks. To test the effectiveness of *MISPAG-DM*, we performed experiments using the PPI network of *S. cerevisiae*. Experimental results show that *MISPAG-DM* can discover interesting and biologically meaningful structural patterns in the PPI network of *S. cerevisiae* that do not occur frequently. It is noted that in these patterns that proteins do not only necessarily interact with proteins that perform the same function, they also can interact with other proteins that perform different functions to form biologically meaningful patterns. Some of these patterns, for example, are found to correspond to protein complexes known to be in *S. cerevisiae*. The interesting patterns that *MISPAG-DM* discovers may potentially contribute to the discovering of biological meaningful patterns that are not yet identified.

Chapter 5

Discovering Interesting Structural Patterns for Graph Attribute Prediction with Applications to Protein Function Prediction

Biomolecules such as genes, proteins, and metabolites are expected to interact with each other in the biological networks of different species. The discovery of interesting interaction patterns can be correlated with the topological entities and functional role of the network and the distortion of the interactions may lead to the development of certain diseases [150, 151, 152]. Determining the functions of genes is an important problem to study for understanding the molecular and biochemical processes that sustain health and cause disease. On average, there are 70% of the genes in a genome are poorly known or with no known functions [91]. However, it should be noted that each species contains thousands to ten thousands of biomolecules, and experimentally determining their functions is an expensive process. Several computation approaches [4, 21, 133] have been proposed to predict molecular function of biomolecules with labeled networks. Most constituent biomolecules are involved in multiple cellular processes and performed more than one molecular function. However, the existing methods are developed to tackle single-attribute network graphs where each vertex is used to represent a single attribute about a biomolecule and each edge about the existence of interaction. These

additional attributes are currently not taken into consideration even though they can be very useful. In order to discover interesting association patterns in biological networks, it is necessary to represent biological networks as multiple-attribute graphs so that more information about the attributes of the constituent biomoleules can be considered during the mining process. Many approaches have been developed to predict functions of the un-annotated genes from the gene interaction networks using the neighborhood concept, and they can be categorized into two types: direct neighborhood, and shared neighborhood.

5.1 Direct Neighborhood

The most straightforward approach of the direct neighborhood is the neighbor counting proposed by Schwikowski et al. [4]. Its scoring function $f_x(p)$ of a gene p with function x is calculated based on the frequency of its occurrence in the interaction neighbors (also called direct neighbors) N_p of p.

$$f_x(p) = \sum_{n \in Np} \delta(n,x) \text{ where} \tag{11}$$

$$\delta(n,x) = \begin{cases} 1 & \text{if } n \text{ has function } x \\ 0 & \text{otherwise} \end{cases}$$

If function y has the largest score $f_y(p)$ among other functions, it means y occurs most frequently in the direct neighbors of p, and the function y will be

assigned to p. In [4], it has been demonstrated that by assigning the top three frequent neighborhood functions to the gene products in the gene interaction network of yeast, the neighbor counting approach can correctly predict 72% of the 1393 characterized gene products with at least one neighbor of known function.

However, the neighbor counting approach, that only considers the frequency of a function appears in the direct neighbors, will ignore the frequency distribution of certain functions annotated in the other genes. If function x and y are both appeared the same number of times in the direct neighbors of gene p, the neighbor counting approach will assign the same score to both functions.

Instead of using the frequency as a scoring function, Hishigaki et al. [21] proposed to use the chi-square statistical method [46] as scoring function to perform the prediction. The statistical measure calculates the deviation of the observed occurrence of function x in the direct neighbors of biomolecule p from its expected occurrence. The equation of the chi-square scoring function is:

$$C_x(p) = \frac{(f_x(p) - e_x(p))^2}{e_x(p)} \tag{12}$$

where $e_x(p)$ is the expected occurrence of biomolecules with function x among the direct interacting neighbors of p. The frequency of x in the direct neighbors of p is compared with the expected occurrence across the whole network. Although the functions assigned by the chi-square approach are more significant than the neighbor

counting approach, its accuracy will be dropped if too many neighbors are considered. The problem may due to the noise and redundancies in the network.

5.2 Shared-Neighborhood

Based on the observation that biomolecules that share a number of neighbors are more likely to have a function in common, the concept of shared neighborhood is introduced. Samanta et al. [47] proposed the concept of shared neighborhood to define the edge in a network graph. In this approach, an edge will be formed if two genes are sharing some neighbors, and the significance value of the edge is calculated by hypergeometric p-value [48]. The p-value shows the degree of likelihood that the two genes share neighbors by chance in a network. A smaller p-value reflects the observation is more likely to be biological significant.

The concept of shared neighborhood is also utilized in PRODISTIN [22] which calculates the distance between two genes by using the Czekanowski-Dice distance. With this approach, new functions were predicted for 37 genes in [49], and 12 of them are novel prediction. It is believed that if two genes have a large number of shared neighbors, the likelihood of these two genes sharing a function becomes significantly higher. However, it should also be noted that not all interactions are taken place between biomolecules with common function. It was discovered that 35% of the interactions in yeast PPI network were between proteins with no common functional annotation [4]. Instead of finding the related function from either direct or

shared neighborhood, we aim to mine the dependency between the functional annotations of genes from a hybrid network graph.

5.3 Hybrid-Neighborhood

Here we transform the original network into a hybrid network graph that contains both direct and shared neighbors. As mentioned previously in Section 3.1, the connectivity of biomolecules in the biological interaction network follows a power-law distribution instead of the exponential distribution expected from random networks, so we used p-value to define the degree of significance between two biomolecules in a biological network. The hybrid network graph contains both direct and shared neighborhood relations, and the relations of shared neighborhood are quantified with the degree of significance.

The multiple interactions between biomolecules in biological networks are represented as a network graph *G* that composes of a set of sub-networks **S** which is responsible for certain function and behavior. There are various types of biological interaction networks in the cell, including protein-protein interaction, metabolic, and gene regulatory networks. They can be represented as undirected network graph G = (**V**, **E**) where **V** refers to a set of biomolecules and **E** refers to a set of relationships between the vertices in **V**. Different topologies affect specific network properties. It is important to understand and model the topological and dynamic properties of various biological networks in a quantifiable manner.

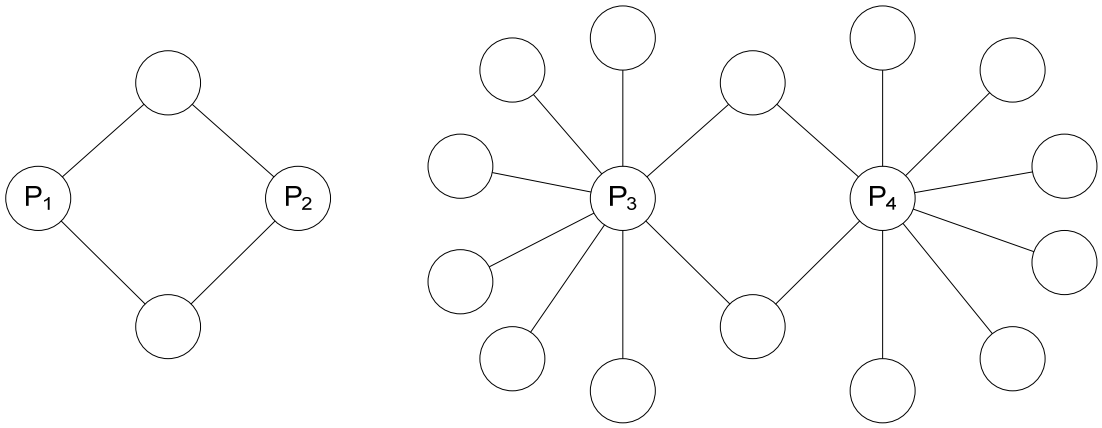
It should be noted that each species contains thousands to ten thousands of biomolecules such as genes, determining gene function experimentally is an expensive process. Several computation approaches have been proposed to predict the gene function with the labeled network. A straightforward method is neighbor counting approach [4]. As genes with similar function tend to cluster together, the function of an un-annotated gene is predicted with its direct neighbors. The most dominant functional GO term of the neighborhood will be assigned to the un-annotated gene. However, a gene is likely to have multiple functions and related genes may share a number of common neighbors instead of directly interact [63]. Another version of neighborhood-based approach [64] is proposed by Bader et al. to consider the gene pairs with common neighbors. Unlike the traditional direct neighbor approach that the accuracy usually decreases as the number of neighbor increases, the common neighbor approach attains a relatively stable level of accuracy. However, it should also be noted that not all interactions are taken place between biomolecules with common function. It was discovered that 35% of the interactions in yeast PPI network were between proteins with no common functional annotation [4]. Instead of finding the related function from either direct or shared neighborhood, we aim to mine the dependency between the functional annotations of genes from a hybrid network graph.

By considering the graph properties of biological network, we transform the original network G into a hybrid network graph G*'* that contains both direct and shared neighbors. We define an attributed graph as an ordered pair G = (**V**, **E**) where

V = $\{ v_1 , ..., v_{m_v} \}$ is a set of attributed vertices and **E**=$\{ e_1 , ..., e_{m_e} \}$ is a set of attributed edges. For each pair of vertices v and v_i, an edge e will be added to the graph G' if they share at least one common neighbor. The weight of that edge is indicated by the degree of significance Q.

As described earlier by Von Mering et al. [65] and Jeong et al., [66], the connectivity of biomolecules in the biological interaction network follows a power-law distribution instead of the exponential distribution expected from random networks. Within this distribution, Bader et al. discovered that essential proteins show a higher level of connectivity ($\overline{k}$ = 10.7) than nonessential proteins ($\overline{k}$ = 5.0) [64]. Hence, we use p-value [47] to define the degree of significance between two biomolecules in biological network.

Let us consider an example in *Figure* 20 that shows two sub-networks from a PPI network with 50 proteins. The relationship between the labeled proteins in Case 1 (*Figure* 20a) is more significant than Case 2 (*Figure* 20b). The reason is that protein P_1 and P_2 in Case 1, each has only two neighbors, share both of these neighbors; whereas the protein P_3 and P_4 in Case 2, each has eight neighbors, share only two of them. This can be proved by comparing the degree of significance of (P_1, P_2), and (P_3, P_4).



(a) Case 1 (b) Case 2

Figure 20 Two protein sub-networks with two shared neighbors

Assume we have two proteins, A and B, the degree of significance Q is:

$$Q(\text{A, B}) = -\log(\text{p-value}) \tag{13}$$

The p-value is the probability of obtaining a result at least as extreme as the one that was actually observed. For the ease of illustration, we take a negative log to obtain a positive value for Q, so that the larger the value of Q, the more significant the connection. The equation of p-value $P(N, n_{(A)}, n_{(B)}, m_{(AB)})$ is given as:

$$P(N, n_{(A)}, n_{(B)}, m_{(AB)}) = \frac{\binom{N}{m_{(AB)}}\binom{N - m_{(AB)}}{n_{(A)} - m_{(AB)}}\binom{N - n_{(A)}}{n_{(B)} - m_{(AB)}}}{\binom{N}{n_{(A)}}\binom{N}{n_{(B)}}}$$

$$= \frac{(N - n_{(A)})!(N - n_{(B)})!n_{(A)}!n_{(B)}!}{N!m_{(AB)}!(n_{(A)} - m_{(AB)})!(n_{(B)} - m_{(AB)})!(N - n_{(A)} - n_{(B)} + m_{(AB)})!} \tag{14}$$

where

N : the total number of proteins

$n_{(A)}$: the number of neighbors of protein A

$n_{(B)}$: the number of neighbors of protein B

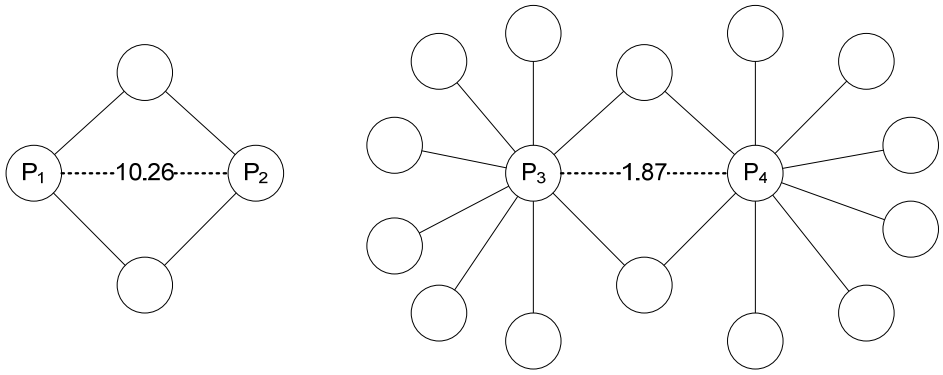
$m_{(AB)}$: the number of shared neighbors that interact with both protein A and B

Their degrees of significance are calculated as follows, and the result proves that the relation between the specified proteins in Case 1 is more significance than Case 2.

$Q(P_1, P_2) = P(50, 2, 2, 2) = 10.26$

$Q(P_3, P_4) = P(50, 8, 8, 2) = 1.87$

Figure 21 shows the two hybrid network graphs for the cases in *Figure* 4. The extra edges (dotted line) are added to each of the network with the degree of significance as the weight of the edge.



(a) Case 1 (b) Case 2

Figure 21 Comparison of two hybrid networks

To evaluate how the degree of significance influences the functional associations between the neighbors of a gene, we select a gene YDR041W (*Figure* 22) from the yeast interaction database for illustration. The gene YDR041W is directly interacted with three genes as direct neighbors, and indirectly connected with 40 genes in which each of them has a certain number of shared neighbors with YDR041W. In order to predict the function of the query gene YDR041W, we need to analyze the associations between the genes by *MISPAG-PF* based on the interestingness measure defined in Chapter 3.

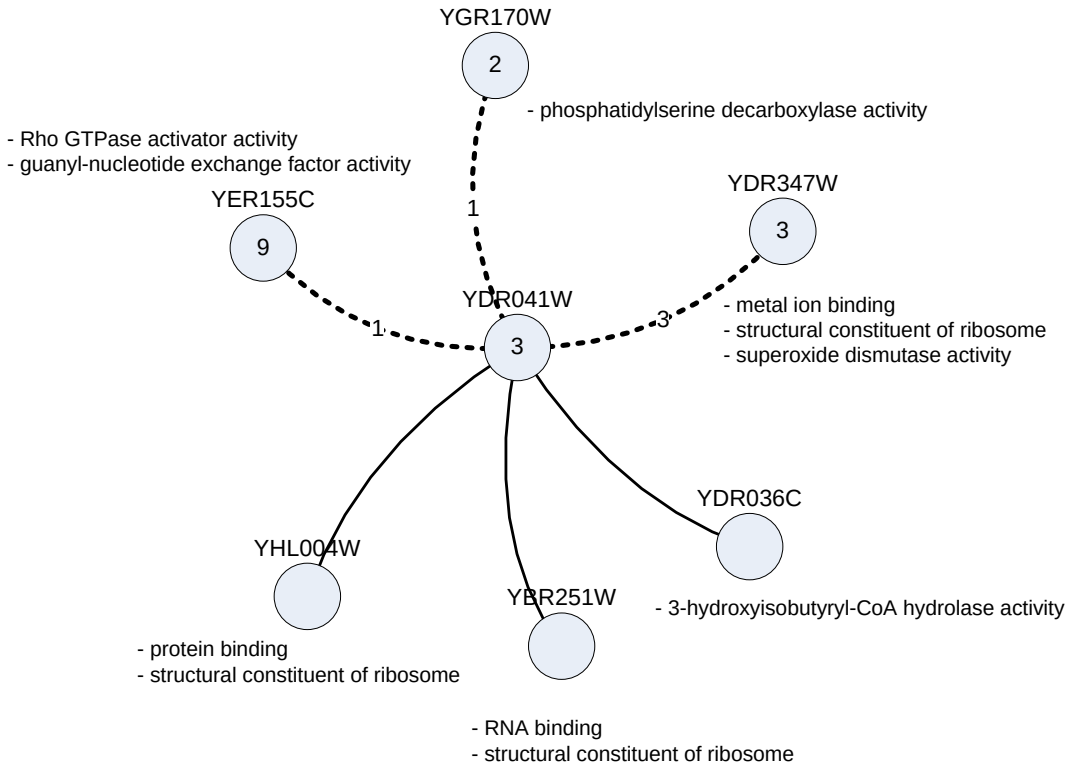


Figure 22 Direct and shared neighbors of a yeast gene YDR041W

5.4 Mining Interesting Association Patterns

Each neighboring gene has one or more molecular functions, for example, YHL004W has two functions: protein binding and structural constituent of ribosome, whereas YGR170W has only one function: phosphatidylserine decarboxylase activity. By considering the interesting function associations of the neighboring genes, we can predict the function of the query gene.

In order to identify the qualified association patterns that are interesting and useful, we proposed *MIAPMAG-PF* which can be used to effectively discover interesting patterns which can be applied to predict the molecular function of biomolecules. To deal with the problem of representing multiple attributes in a graph, the vertices and edges in *MIAPMAG-PF* are allowed to have multiple attributes A_1, A_2, …, A_q, and each vertex v_i can have multiple attribute values a_{ij}^1, a_{ij}^a, …, a_{ij}^A, corresponding to each attribute j. The definition of multiple-attribute (MA) graph will be described in details in Chapter 6.

To discover interesting association from network graphs, first of all, each candidate edge is verified with the interestingness measure to determine if it appears more frequently in one class than the other, and screen out the uninteresting ones. *MIAPMAG-PF* makes use of a test statistics similar to Chapter 3 to define the adjusted residual value for distinguishing interesting associations from the uninteresting ones. These interesting patterns can be used for the purpose of function prediction.

Once the set of candidate associations, $\mathbf{S}^{(p)}$, $p = 1, \ldots, P$, are discovered for each of $\mathfrak{G}^{(p)}$, $p = 1, \ldots, P$, respectively, the probability that a graph, G, is in $\mathfrak{G}^{(p)}$ $p \in \{1, \ldots, P\}$, given that G is characterized by a candidate association, $S_j^{(p)} \in \mathbf{S}^{(p)}$, $j \in \{1, \ldots, m_p\}$ can be determined as Equation (1). If $\Pr(G \in \mathfrak{G}^{(p)} \mid$ G is characterized by $S_j^{(p)})$ is not much different from $\Pr(G \in \mathfrak{G}^{(p)})$, then $S_j^{(p)}$ should not be considered very interesting in determining if G should be classified into $\mathfrak{G}^{(p)}$. Otherwise, $S_j^{(p)}$ can be very interesting.

To objectively determine if the two probabilities are different, we make use of the adjusted residual value, d_{ji} which is defined as Equation (2). If $|d_{jp}| > 1.96$, we can conclude that the difference between $\Pr(G \in \mathfrak{G}^{(p)} \mid G$ is characterized by $S_j^{(p)})$ is significantly different from $\Pr(G \in \mathfrak{G}^{(p)})$ and therefore the association $S_j^{(p)}$ is interesting and useful for classification. If $d_{jp} > +1.96$, it implies that the presence of the candidate association $S_j^{(p)}$ in a graph G provides evidence supporting G to be classified into $\mathfrak{G}^{(p)}$ otherwise if $d_{ji} < -1.96$, it implies that the presence of the candidate association $S_j^{(p)}$ provides negative evidence against G to be classified into $\mathfrak{G}^{(p)}$. In either case, $S_j^{(p)}$, can be qualified as an *interesting* association pattern.

MISPAG-PF screens each set of candidate association, $\mathbf{S}^{(p)} = \{ S_1^{(p)}, \ldots, S_s^{(p)}, \ldots, S_{m_p}^{(p)} \}$, $p = 1, \ldots, P$, to retain only those who are interesting. The set of

interesting candidate associations discovered for each of $\mathfrak{G}^{(1)}$, …, $\mathfrak{G}^{(p)}$, …,$\mathfrak{G}^{(P)}$ respectively is denoted as $\mathbf{S}'^{(p)}=\{ S_1'^{(p)}, \ldots, S_j'^{(p)}, \ldots, S_{m_p'}'^{(p)}\}$, $p = 1, \ldots, P$, and $m_p' < m_p$. Rather than a random collection of vertices and edges, these filtered association rules may have significant biological meaning as they may represent the inter-relationship between functional proteins. *MIAPMAG-PF* has the advantage that it can significantly reduce the number of interaction pairs by filtering those irrelevant one. As *MIAPMAG-PF* does not require any user-defined threshold, such as the use of a support and confidence measure, which can only be obtained by trial-and-error, it can discover associations with relatively low frequency but are useful in function prediction for each network class.

The top three interesting function associations of the query gene with direct neighbors and shared neighbors are listed in *Table* 12 and *Table* 13 respectively. We can conclude that the function of YDR041W is structural constituent of ribosome as it is the most significant associations (bolded) in both direct and shared neighborhood approach.

Table 12 Direct neighbors and function associations related to YDR041W

Gene	**Function**	**Predicted Function** (adjusted residual value)
YHL004W	Protein binding	Peptidase activity (6.48)
		RNA splicing factor activity (4.63)
		ATP_binding (4.25)
	Structural constituent of	**Structural constituent of ribosome (24.93)**

	ribosome	NADH dehydrogenase (ubiquinone) activity (15.46)
		Superoxide dismutase activity (10.11)
YBR251W	RNA binding	3^ 5^ exoribonuclease activity (8.76)
		4 iron, 4 sulfur cluster binding (5.36)
		Structural constituent of ribosome (2.34)
	Structural constituent of ribosome	**Structural constituent of ribosome (24.93)**
		NADH dehydrogenase (ubiquinone) activity (15.46)
		Superoxide dismutase activity (10.11)
YDR036C	3-hydroxyisobutyryl-CoA hydrolase activity (hydrolase activity)	**Structural constituent of ribosome (6.71)**
		RNA_polymerase_II_transcription_elongation (5.79)
		carbamoyl-phosphate synthase (5.47)

Table 13 Shared neighbors and function associations related to YDR041W

Gene	**Function**	**Predicted Function** (adjusted residual value)
YER155C	Rho GTPase activator activity	**Structural constituent of ribosome (8.55)**
		Endoplasmic reticulum signal peptide binding (5.33)
		RNA binding (5.09)
	Guanyl-nucleotide exchange factor activity	**Structural constituent of ribosome (5.32)**
		Translation initiation factor activity (3.42)
		RNA binding (2.55)
YGR170W	Phosphatidylserine decarboxylase activity	**Structural constituent of ribosome (8.09)**
		Superoxide dismutase activity (5.74)
		gamma-tubulin binding (4.39)
YDR347W	Metal ion binding	**Structural constituent of ribosome (10.70)**
		gamma-tubulin binding (4.96)

		5^-3^ exoribonuclease activity (3.41)
	Structural constituent of ribosome	Superoxide dismutase activity (22.41)
		gamma-tubulin binding (15.44)
		Ribonuclease III activity (10.76)
	Superoxide dismutase activity	**Structural constituent of ribosome (22.41)**
		gamma-tubulin binding (9.85)
		Phosphatidylserine decarboxylase_activity (5.74)

5.5 Data Description

The gene interaction data is collected from BioGRID database [16] that records the relationships between genes. The interaction reported in BioGRID is direct and physical in nature, and the experimental system definitions indicate the nature of the supporting evidence for an interaction between the two biological units. We selected the experimental data set from the 2.0.20 version of BioGRID for Saccharomyces cerevisiae (yeast) that contains 5,299 yeast genes (vertices) and 82,633 interactions (edges). The reason of choosing this set of data is that the experimental protocols, physiology and metabolism of yeast are well-defined comparing to multi-cellular species. As a uni-cellular species, it is a relatively simple system to study, so there are many biologists have performed various practical experiments and annotated most of the yeast genes. The percentage of unknown function of yeast gene is relatively low, there is only 7% of genes are un-annotated.

In order to predict the gene function more precisely, we adopt a hybrid neighbor approach that consider the dependency between both direct neighbors and shared neighbors in a gene network. To model a hybrid network graph as described in Section 3.1.2, each vertex represents a gene, and edges represent both the direct interactions and shared neighborhood relations.

5.6 Experiments and Results

We model the yeast network with the functional annotation data in the Functional Catalogue (FunCat) [14] in MIPS, and the genes are functional annotated in GO database. The MIPS FunCat is an annotation scheme for the functional description by mapping of GO annotation and the literature. It consists of 28 main functional categories which covers the general fields like cellular transport, metabolism and cellular communication/signal transduction. Within the 5,299 yeast genes, 76% of them are functionally annotated, and some of the genes may belong to more than one functional class. Before performing the gene function prediction, we label these genes in the yeast network with these GO terms.

MISPAG-PF algorithm is applied to discover the relationships between the gene functions of yeast from the hybrid network graph. In the following, gene function prediction is formalized as a classification problem. By performing a 10-fold cross-validation to partition the annotated subset into training and test set, we can evaluate the performance of different algorithms. During the training stage, *MISPAG-PF* will discover association rules of direct interacting genes and shared

neighborhood genes from the training set, and each of the rules are characterized with an interestingness measure.

With the discovered association rules, we can predict the GO function of the genes in the test set. As the interacting associations between different functions are varied, instead of assigning the common function of the direct neighbors or shared-neighbors to the target gene, a better way is to define the relatedness of the functions by using the total interestingness measure as mentioned in Section 3.6. The top 50 direct associations and 50 shared neighborhood associations that are discovered in the training stage are listed in *Table* 14 and *Table* 15 respectively. In these *Table*s, each row represents two molecular functions with the association significance that denoted by the adjusted residual value *d* that defined in Section 3.4.

Table 14 Top 50 direct associations between the GO functions in the gene interaction data of yeast

Molecular function 1	**Molecular function 2**	***d***
spermine_synthase_activity	spermidine_synthase_activity	173.7
alpha-glucosidase_activity	alpha-glucosidase_activity	57.9
ribonuclease_P_activity	kinesin_binding	48.2
ribonuclease_P_activity	ribonuclease_MRP_activity	48.2
dolichyl-diphosphooligosaccharide-protein_g...	adenosylmethionine-8-amino-7-oxononanoate_t...	45.3
soluble_NSF_attachment_protein_activity	SNARE_binding	40.9
ribonuclease_P_activity	ribonuclease_P_activity	40.0
actin_binding	actin_binding	39.7
amino_acid_binding	acid_phosphatase_activity	38.8
phosphoglycerate_dehydrogenase_activity	acid_phosphatase_activity	38.8
sterol_carrier_activity	FFAT_motif_binding	38.6
RNA_polymerase_III_transcription_factor_act...	RNA_polymerase_III_transcription_factor_act...	38.5

alpha-glucosidase_activity	cation_binding	35.4
amino_acid_binding	amino_acid_binding	34.7
amino_acid_binding	phosphoglycerate_dehydrogenase_activity	34.7
sterol_carrier_activity	phosphoinositide_binding	32.8
RNA_polymerase_II_transcription_mediator_ac...	RNA_polymerase_II_transcription_mediator_ac...	31.9
ribonuclease_P_activity	ribonuclease_activity	27.8
transcription_regulator_activity	chaperone_activator_activity	26.8
transcription_regulator_activity	cholesterol_binding	26.8
transcription_regulator_activity	nitric-oxide_synthase_binding	26.8
transcription_regulator_activity	protease_activator_activity	26.8
transcription_cofactor_activity	3-beta-hydroxy-delta5-steroid_dehydrogenase...	25.0
transcription_cofactor_activity	C-3_sterol_dehydrogenase_(C-4_sterol_decarb...	25.0
transcription_cofactor_activity	sterol-4-alpha-carboxylate_3-dehydrogenase_...	25.0
structural_constituent_of_ribosome	structural_constituent_of_ribosome	24.9
SNAP_receptor_activity	SNARE_binding	23.6
acetyl-CoA_carboxylase_activity	enzyme_activator_activity	23.4
enzyme_activator_activity	biotin_carboxylase_activity	23.4
citrate_(Si)-synthase_activity	L-iditol_2-dehydrogenase_activity	23.0
phospholipid_binding	GDP_binding	22.4
protein-N(PI)-phosphohistidine-sugar_phosph...	inorganic_diphosphatase_activity	21.8
phosphoenolpyruvate-protein_phosphotransfer...	inorganic_diphosphatase_activity	21.8
inorganic_diphosphatase_activity	sugar:hydrogen_symporter_activity	21.8
cation_binding	cation_binding	21.7
histone_binding	transition_metal_ion_binding	21.0
histone_binding	ribonucleoside-diphosphate_reductase_activity	21.0
hormone_activity	alpha-mannosidase_activity	20.9
protein_domain_specific_binding	alpha,alpha-trehalase_activity	20.3
DNA-directed_RNA_polymerase_activity	two-component_sensor_activity	20.1
phosphatidylinositol_binding	kynurenine-oxoglutarate_transaminase_activity	19.9
enzyme_activator_activity	phosphoacetylglucosamine_mutase_activity	19.8
two-component_sensor_activity	ubiquitin_protein_ligase_binding	19.6
carboxylesterase_activity	importin-alpha_export_receptor_activity	19.4
transcription_factor_binding	receptor_activity	19.4
transcription_regulator_activity	chaperone_binding	18.9
hydroxymethylglutaryl-CoA_synthase_activity	RNA_polymerase_I_transcription_factor_activity	18.8
formate_dehydrogenase_activity	CTP_synthase_activity	18.7
biotin_binding	enzyme_activator_activity	18.5
phosphoribosylformylglycinamidine_cyclo-lig...	RNA_polymerase_I_transcription_termination_...	18.3

Table 15 Top 50 shared-neighborhood associations between the GO functions in the gene interaction data of yeast

Molecular function 1	**Molecular function 2**	***d***
amino_acid_binding	acid_phosphatase_activity	225.0
phosphoglycerate_dehydrogenase_activity	acid_phosphatase_activity	225.0
SNAP_receptor_activity	soluble_NSF_attachment_protein_activity	83.7
3^,5^-cyclic-nucleotide_phosphodiesterase_a...	phospholipid-translocating_ATPase_activity	71.1
kinesin_binding	ribonuclease_P_activity	71.1
ribonuclease_P_activity	ribonuclease_activity	71.1
ciliary_neurotrophic_factor_receptor_binding	S-adenosylmethionine-dependent_methyltransf...	56.7
interleukin-6_receptor_binding	S-adenosylmethionine-dependent_methyltransf...	56.7
ribonuclease_P_activity	ribonuclease_MRP_activity	47.4
kinesin_binding	ribonuclease_MRP_activity	45.0
kinesin_binding	ribonuclease_activity	45.0
ribonuclease_MRP_activity	ribonuclease_activity	45.0
ribonuclease_P_activity	ribonuclease_P_activity	37.4
dolichyl-diphosphooligosaccharide-protein_g...	adenosylmethionine-8-amino-7-oxononanoate_t...	35.0
3^,5^-cyclic-nucleotide_phosphodiesterase_a...	ATPase_activity,_coupled_to_transmembrane_m...	30.0
chaperone_activator_activity	cholesterol_binding	27.3
chaperone_activator_activity	nitric-oxide_synthase_binding	27.3
chaperone_activator_activity	protease_activator_activity	27.3
alanine-glyoxylate_transaminase_activity	cation_binding	25.8
alanine-glyoxylate_transaminase_activity	chitin_binding	25.8
alanine-glyoxylate_transaminase_activity	glucan_1,3-beta-glucosidase_activity	25.8
alanine-glyoxylate_transaminase_activity	glucan_endo-1,3-beta-D-glucosidase_activity	25.8
phospholipase_D_activity	trans-2-enoyl-CoA_reductase_(NADPH)_activity	25.4
endoplasmic_reticulum_signal_peptide_binding	Ran_GTPase_activator_activity	25.3
sterol_carrier_activity	tubulin_binding	25.3
ribonucleoside-diphosphate_reductase_activity	transition_metal_ion_binding	23.9
chaperone_binding	cholesterol_binding	23.4
chaperone_binding	nitric-oxide_synthase_binding	23.4
chaperone_binding	protease_activator_activity	23.4
xenobiotic-transporting_ATPase_activity	phospholipid-translocating_ATPase_activity	23.4
structural_constituent_of_ribosome	superoxide_dismutase_activity	22.4
tubulin_binding	tubulin_binding	22.4
RNA_polymerase_III_transcription_factor_act...	RNA_polymerase_III_transcription_factor_act...	22.1
ciliary_neurotrophic_factor_receptor_activity	interleukin-1_binding	21.9
ciliary_neurotrophic_factor_receptor_activity	interleukin-8_binding	21.9

ciliary_neurotrophic_factor_receptor_activity	tumor_necrosis_factor_binding	21.9
ciliary_neurotrophic_factor_receptor_activity	wide-spectrum_protease_inhibitor_activity	21.9
arsenite_transmembrane-transporting_ATPase_...	FK506_binding	21.7
phosphatidylinositol_binding	kynurenine-oxoglutarate_transaminase_activity	20.5
dolichyl-diphosphooligosaccharide-protein_g...	protein_N-terminus_binding	20.1
RNA_polymerase_I_transcription_factor_activity	endonuclease_activity	19.7
alpha,alpha-trehalase_activity	alpha,alpha-trehalase_activity	18.7
transcription_corepressor_activity	transition_metal_ion_binding	18.6
xenobiotic-transporting_ATPase_activity	3^,5^-cyclic-nucleotide_phosphodiesterase_a...	18.4
peptidase_activity	proteasome_activator_activity	18.2
protein_binding	RNA_binding	17.7
hydrogen_ion_transporting_ATPase_activity,_...	endodeoxyribonuclease_activity	17.4
phosphoenolpyruvate-protein_phosphotransfer...	inorganic_diphosphatase_activity	17.3
protein-N(PI)-phosphohistidine-sugar_phosph...	inorganic_diphosphatase_activity	17.3
sugar:hydrogen_symporter_activity	inorganic_diphosphatase_activity	17.3

As we observed from the direct associations of the gene interaction network of yeast, only 10% of them are between genes with common function such as alpha-glucosidase activity (d = 57.9), ribonuclease P activity (d = 40.0), and actin binding (39.7). Instead, many interesting associations are formed between genes with different functions, for example, a gene with the molecular function of amino acid binding is more likely to interact with the gene with another molecular function of acid phosphate activity (d = 38.8) than the gene with the same function (d = 34.7). The case is the same for those genes in the shared neighborhood associations. These associations are the interesting rules that are essential to predict gene function.

Within the shared neighborhood associations of yeast genes, *Figure* 23 shows that over 70% of the associations are between genes with one shared neighbor, and only less than 1% associations have over nine shared neighbors.

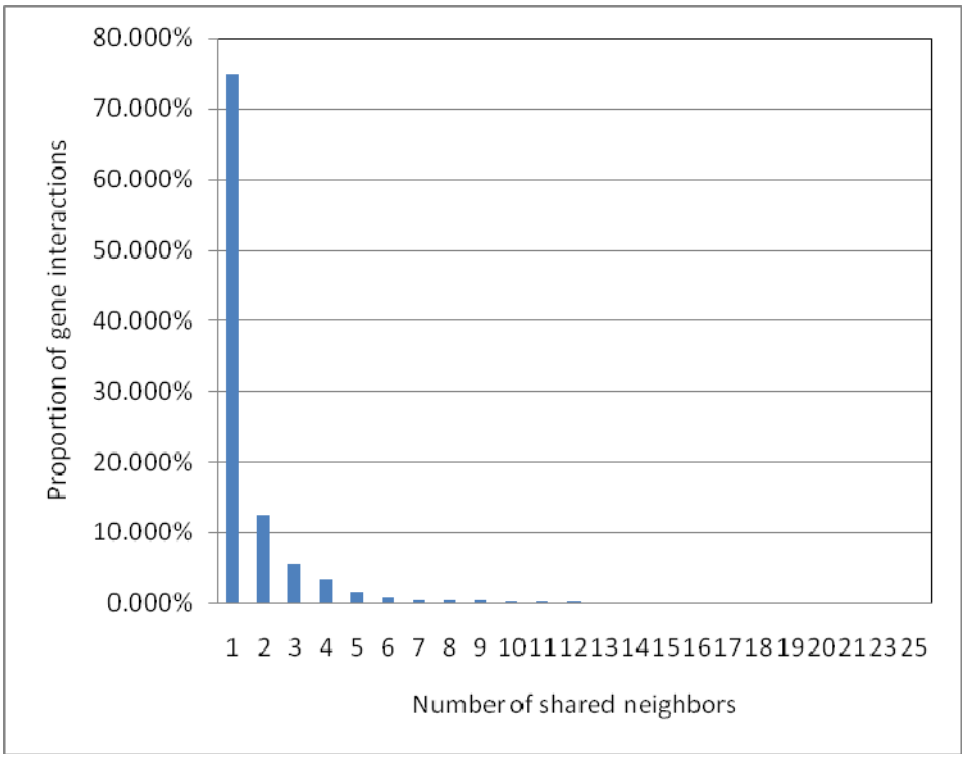


Figure 23 Distribution of gene interactions with different number of shared neighbors

In order to verify the factor that influence the prediction accuracy of gene function, the average matching score is plotted against the number of shared neighbors in *Figure* 24, it is observed that the prediction accuracy is not governed by the number of shared neighbors. Instead, the matching score is higher for those with higher degree of significance as observed in *Figure* 25.

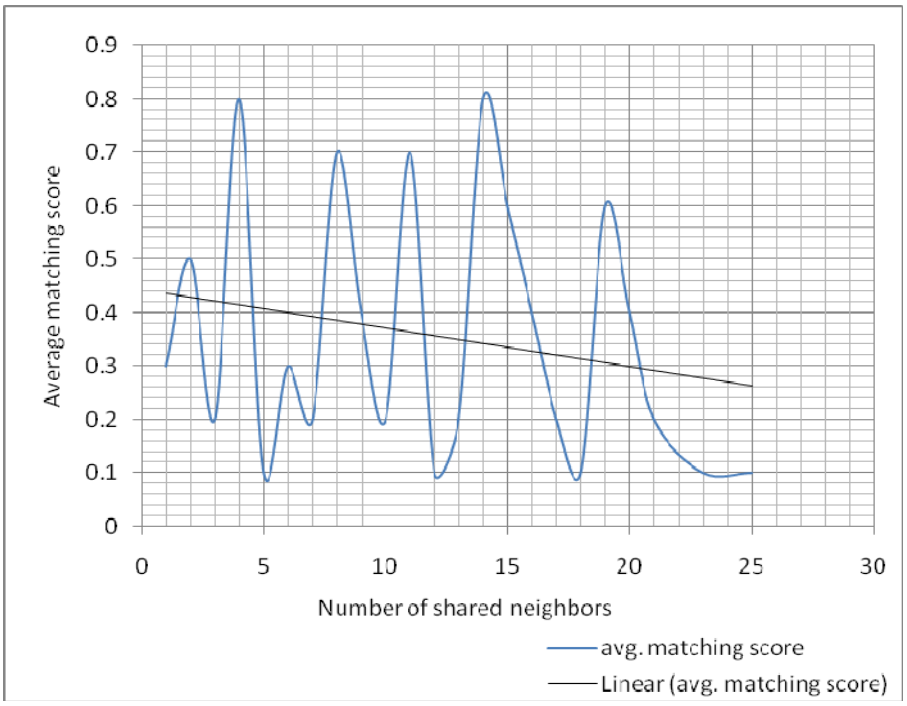


Figure 24 Average matching score of different number of shared neighbors

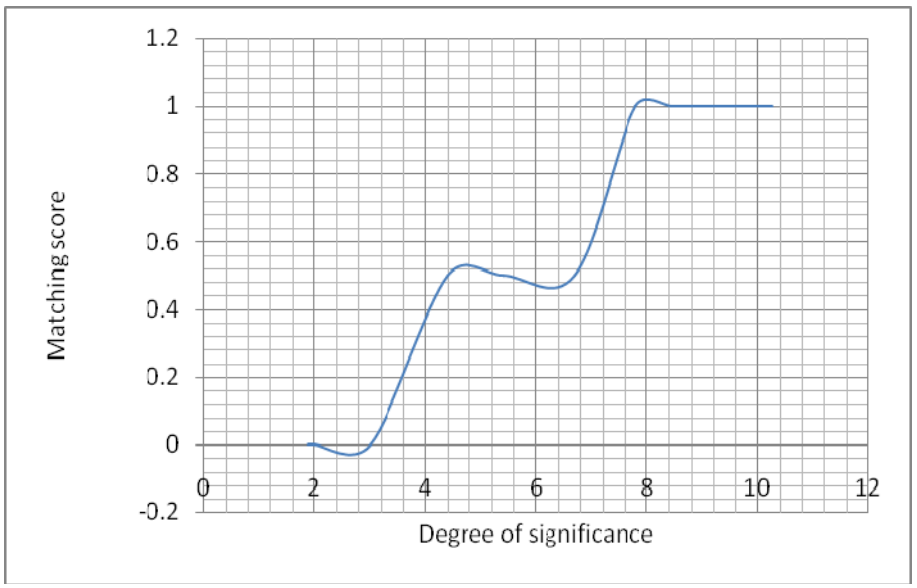


Figure 25 Matching score of different degree of significance

The yeast network is modeled as a hybrid network graph and its edges are calculated with the degree of significance Q (Section 3.1.2) to filter those edges with $Q<6$. *MISPAG-PF* is compared against three other approaches including neighbor counting [4], chi-square statistical method [21] and shared-neighborhood approach [22]. We use the AUC in ROC analysis for evaluation, and the results are given in *Table* 16. The detailed description of ROC analysis can be referred to Section 5.2. The experimental results show that *MISPAG-PF* has the highest accuracy (AUC) when comparing to the other prediction algorithms with the hybrid neighboring approach. It performs better than neighbor counting, chi-square, and PRODISTIN by 1.24, 1.15, and 1.09 times.

Table 16 The prediction results over the gene interaction data of yeast

Methods	**AUC**
MISPAG-PF	0.83
Neighbor counting	0.67
Chi-square	0.72
PRODISTIN	0.76

It should be noted that not all interactions are taken place between genes with common function. It was discovered that 35% of the interactions in the gene interaction network of yeast were between genes with no common functional annotation [4]. The experimental results show that *MISPAG-PF* is an effective algorithm over the other algorithms to discover interesting subgraphs for predicting

gene function from gene interaction network. It used an interestingness measure to mine the dependency between the functional annotations of genes, instead of finding the common function for prediction.

5.7 Summary

In this chapter we have shown that our proposed algorithm improves upon previous methods of direct and shared neighborhood for the task of function prediction in the gene interaction data of yeast, and the extensions of this approach to other species are straightforward.

The method of hybrid network approach is useful to achieve higher prediction performance by two major characteristics. Firstly, by combining the advantages of direct and shared neighborhood approaches, the neighborhood relations are enriched and quantified by the degree of significance. Secondly, the function prediction is performed by calculating the interestingness measure of the gene pairs instead of counting the commonly occurred function in the neighbors. This helps to solve the problem that some gene may direct or indirect interact with related genes instead of common genes.

We believe that as the prediction task becomes harder when analyzing interspecies interactions, the need for methods that can accommodate high levels of missing values and are directly interpretable increases. The next step will be to apply our method to interaction prediction tasks related to important types of disease

related genes where missing values and the small number of positive examples are the major obstacles in obtaining an accurate gene function prediction.

Chapter 6
Discovering Interesting Structural Patterns for Graph Classification with Applications to PPI Network Classification

Biological network involves interaction relations between individual biological components. The study of the structural interaction in biological networks is an important research area in recent years because the biological networks contain a rich amount of information that helps to understand the organization, interactions and functions of the interacting units. The groups of interacting biomolecules potentially share common functions within each network class, so they are potentially useful to perform network classification that predicts the class of unknown biological networks. The problem of network classification typically involves finding a classifier to classify a set of networks or sub-networks into different predefined classes. A set of networks is then used as the training set to construct a classifier. Using the classifier, a network that is not originally in the training set can be classified.

To address the problem of mining patterns from biological networks among different species is a challenging task. Mining the complex structure of biological networks that involves thousands of interactions between thousands of proteins with multiple attributes is regarded as a graph mining problem that related to the NP-hard

isomorphism problem. Besides, the nature of biological network data is usually incomplete. It consists of many functionally unclassified biomolecules. For example, the PPI network of a fruit fly contains over 80% of proteins with unclassified function.

Traditional classification algorithms assume that the given records are represented in relational database with one-dimensional feature vectors. However, a PPI network is represented as a more complicated structure that in the form of a graph with a set of vertices and edges, so the problem of biological network classification can be modeled as graph classification.

Typically, graph classification is divided into two phases: feature vector identification and classification model generation. Deshpande et al. [50] use the frequent subgraph mining algorithm to define the feature vectors for classification. These feature vectors are subgraphs whose support is greater than or equal to the minimum support threshold σ, and they are fed into a classifier for performing the graph classification. With the feature-based representation, classification techniques such as support vector machine (*SVM*) [54] can be used for the classification task. *SVM* is a machine learning classifier that constructs a separating hyperplane in the n-dimensional space of input data. *SVM* is widely adopted to classify structural data. Cai et al. classified protein sequences with *SVM* [55], and Dobson et al. applied *SVM* to distinguish enzyme from non-enzyme proteins [56]. Much of the recent focus on applying *SVM* in graph application is on how to build efficient and valid kernel functions on graphs. Most of these approaches are usually based on constructing a

feature space with decomposing a graph into subgraphs and counting the number of these subgraphs. As *SVM* is only directly applicable for two-class problems, to deal with multiple class problems, several binary classifiers have to apply separately.

Although there are many efficient and scalable frequent pattern mining algorithms exist for itemset mining and sequence mining, developing efficient and scalable algorithms for structural mining is particularly challenging as subgraph isomorphism is a computationally expensive operation.

The structure of PPI networks is so complex that some network graphs consist of thousands of vertices and ten thousands of edges, and each protein usually perform one or more molecular functions, and each vertex can be annotated by multiple molecular functions in the Gene Ontology. The computation complexity is obviously so high that these apriori-like graph mining algorithms are suffering from the exponential explosion problem. As these algorithms are originally designed for mining graphs with single vertex and edge attribute, feature selection should be used to choose one of the attribute values to represent the protein, or several datasets should be prepared so that each of them contains only one attribute value.

Besides, the frequent subgraphs discovered by the existing graph mining algorithms may not be very useful for network classification. For example, several graph mining algorithms [39, 153, 75] are developed to discover functional interaction patterns from the PPI networks of different living organisms at the molecular level. They modeled the problem as apriori-like graph mining algorithm to find frequently occurring interaction patterns as frequent subgraphs from a PPI

network graph. The discovery of such patterns is useful to identify common sub-networks and understand the structural organization and functions between proteins. However, these algorithms are suffering from the problem of including too many patterns that are frequent but not useful as described in Chapter 3. These subgraphs are only quantified by the frequency, and the frequency level has no additional meaning other than showing the number of occurrence. A different concept should be introduced to discover interesting motifs that can represent different levels of network organization and provide true characterization among different networks. In this chapter, we are going to construct discriminative features for graph classification which could preserve the structural properties of the underlying biological networks.

6.1 Representing PPI Networks in Multiple-Attribute Graphs

To take into considerations these multiple attributes of the proteins when mining PPI networks, we propose to use a representation scheme called multiple-attribute (MA) graph representation. Given a set of N PPI networks each belonging to different species, these networks can be represented as N labeled graphs, $G_1 = G_1(\mathbf{V}_1, \mathbf{E}_1)$, …, $G_k = G_k(\mathbf{V}_k, \mathbf{E}_k)$, …, $G_N = G_N(\mathbf{V}_N, \mathbf{E}_N)$, with vertices in the vertex sets, $\mathbf{V}_k$, $k = 1, \ldots, N$, representing proteins, and edges in the edge sets, $\mathbf{E}_k$, $k = 1, \ldots, N$, representing the interactions between these proteins.

To deal with the problem of representing multiple attributes in a graph, the vertices and edges in *MISPAG-MA* are allowed to have multiple attributes $A_1, A_2, \ldots,$

A_q, and each vertex v_i can have multiple attribute values a_{ij}^1, a_{ij}^a, …, a_{ij}^A, corresponding to each attribute j. The scenario of multiple-attribute graph is described in *Figure* 26 in which a vertex can be labeled by three possible attributes A_1, A_2 and A_3, and an edge can be labeled by two possible attributes A_4 and A_5. The vertices and edges can obtain their attribute values through these attributes.

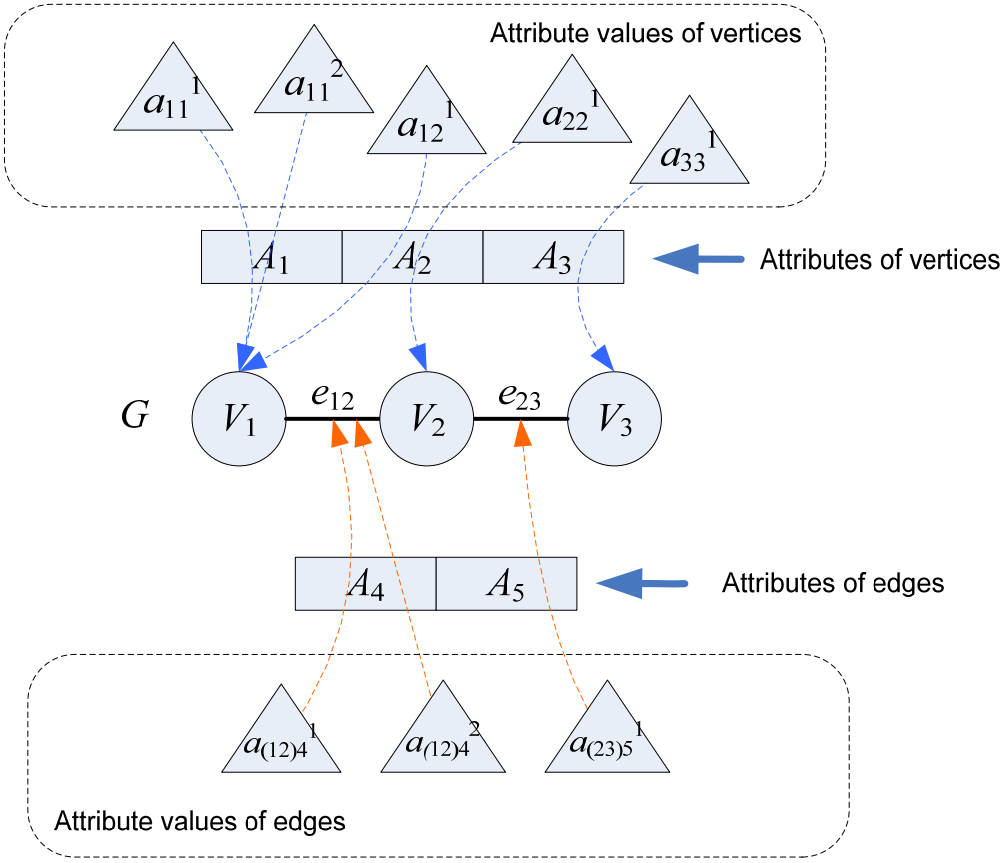


Figure 26 A Multiple-attribute graph example

By means of the object-oriented modeling, we can represent a multiple-attribute graph with its identity, vertex set and edge set. A vertex set contains a list of *Vertex* object, and an edge set contains a list of *Edge* objects. The *Vertex* and *Edge* classes both accept multiple attribute with multiple attribute values, and they are

stored in the variables *vertexType* (and *edgeType)* and vertexValue (and *edgeValue*) respectively. As each attribute type may have more than one attribute values, their values are stored in another class *VertexValue* and *EdgeValue*. With this representation, we can query all possible attribute values of the vertices and edges in the mining process.

As discussed above, for each constituent protein of a PPI network, a number of attributes are known about it and each attribute can usually take on a number of different values. For example, the proteins that make up a PPI network may perform one or more molecular functions, be involved in one or more biological processes, and be located in one or more cellular components. For example, the protein, Myosin light chain 1, with the ID, UniProtKB/Swiss-Prot P53141, in the species of *Saccharomyces cerevisiae* performs five molecular functions (calcium ion binding, identical protein binding, motor activity, myosin II heavy chain binding, and myosin V binding), is involved in three biological processes (cytokinesis, protein localization, and vesicle targeting) and is located in four cellular components (cellular bud neck, cellular bud tip, myosin complex, and vesicle).

For this reason, the graphs used to represent N PPI networks here are MA graphs. To describe such graphs, let us consider $G_k = G_k(\mathbf{V}_k, \mathbf{E}_k)$, $k \in \{1, \ldots, N\}$. Let us represent the vertices of G_k as $\mathbf{V}_k = \{ v_1^k, ..., v_i^k, ..., v_{K_v}^k \}$ and the edges of G_k as

$\mathbf{E}_k = \{ e_1^k, ..., e_i^k, ..., e_{K_e}^k \}$, for each element in $\mathbf{V}_k$ and $\mathbf{E}_k$, a set of attributes is associated with it. The set of attributes associated with a vertex, say, v_i^k, $i \in \{1, ..., K_v\}$

can be represented as $\mathbf{A}_i^{v^k} = \{ A_{i,1}^{v^k}, ..., A_{i,j}^{v^k}, ..., A_{i,I_v}^{v^k} \}$ and each attribute $A_{i,j}^{v^k}$, j = 1, …, I_v, can take on values from *domain*($A_{ij}^{v^k}$)=$\{ a_{i,j,1}^{v^k}, ..., a_{i,j,l}^{v^k}, ..., a_{i,j,J_v}^{v^k} \}$. Similarly, the set of attributes associated with an edge, say, e_i^k, $i \in \{1, ..., K_e\}$ can be represented as $\mathbf{A}_i^{e^k} = \{ A_{i,1}^{e^k}, ..., A_{i,j}^{e^k}, ..., A_{i,I_e}^{e^k} \}$ and each attribute $A_{i,j}^{e^k}$, j = 1, …, I_e, can take on values from *domain*($A_{i,j}^{e^k}$)=$\{ a_{i,j,1}^{e^k}, ..., a_{i,j,l}^{e^k}, ..., a_{i,j,J_e}^{e^k} \}$.

Such a scheme can be used to represent such complex structure from such annotation database as Gene Ontology in UniProt [8]. Given such a set of MA graphs corresponding to a set of PPI networks, a set of attributes are associated with each vertex and edge and each attribute can also be associated with multiple values, and we can discover interesting subgraphs with different degrees of interestingness corresponding to each class of networks.

Given *N* PPI networks represented in *N* multiple attribute graphs as defined above, we use *MISPAG-MA* to effectively discover interesting structural patterns in these graphs. *MISPAG-MA* performs its tasks in several iterative steps. It begins with a first step to select an element from the edge set of a MA graph. The edge, together with the two vertices that it connects to, form a first-order MA pattern. After such a pattern is formed, the attribute values associated with the vertices and the edge in it are then evaluated to determine if they form part of an interesting pattern. The evaluation is based on the use of an objective interestingness measure defined in terms of test statistics defined in terms of the probabilities of occurrences. Based on such a measure, attribute values that are not interesting are removed from the first-

order MA pattern. By repeating such a procedure, all first-order interesting MA patterns can be identified. Such patterns that are connected can then be combined to form second-order MA patterns which consist of two connected edges. The process of screening out uninteresting attribute values can then be repeated for the attribute values associated with the vertices and edges. After screening out of the uninteresting attribute values, the remaining second-order MA patterns can then be searched again to see if they can be connected to form a third order pattern. This process of screening out uninteresting attribute values, forming higher order MA patterns are repeated until no higher order patterns can be formed. We will then have the maximal interesting MA subgraphs representing maximal structural patterns in a PPI network. After such subgraphs are found, a confidence measure can then be computed for each subgraph. This measure can be combined to form an overall total interestingness measure. These interesting patterns together with the confidence measure can be used for the purpose of classifying unknown networks. The general overview of *MISPAG-MA* is given in *Figure* 27.

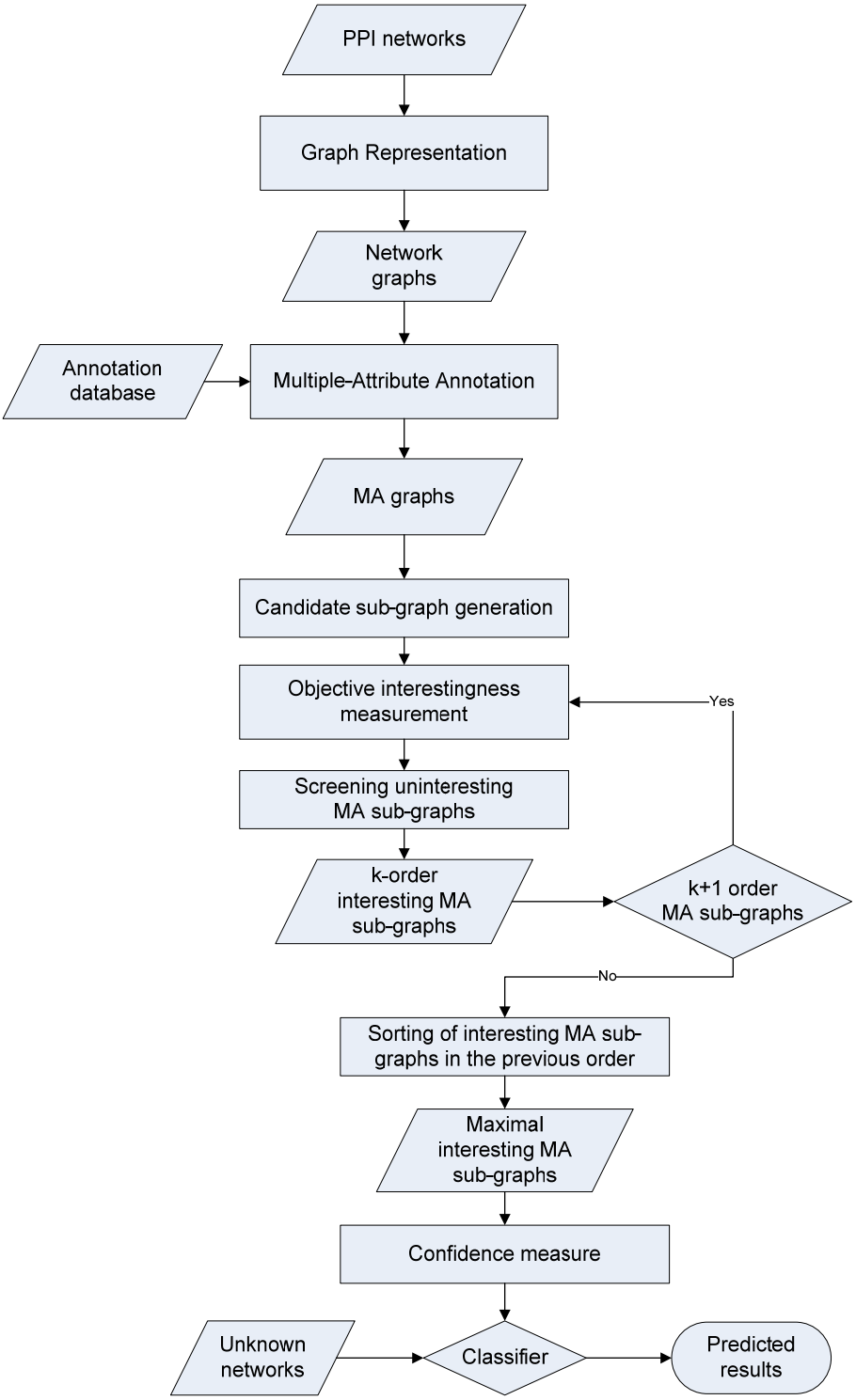


Figure 27 Overview of MISPAG-MA to discover interesting structural patterns in PPI networks

6.2 Screening Out Uninteresting Attribute Values

For a PPI network represented as an MA graph, say, $G_k = G_k(\mathbf{V}_k, \mathbf{E}_k)$, $k \in \{1, \ldots, N\}$ as described above, we can discover interesting structural patterns by following a procedure that consists of several steps. The first step is for *MISPAG-MA* to examine each edge and its two connecting vertices in G_k to determine if it contains interesting patterns to characterize a PPI network to allow it to be distinguished from the others and to allow one to determine if a graph or subgraph represents the PPI network of a particular species. To find such interesting patterns, let us consider each edge $e_i \in \mathbf{E}_k = \{ e_1^k, ..., e_i^k, ..., e_{K_e}^k \}$ and its two connecting vertices v_i and $v_i^{'} \in \mathbf{V}_k = \{ v_1^k, ..., v_i^k, ..., v_{K_v}^k \}$. Other than a single node, such vertex-edge-vertex structure is the smallest subgraph one can find in G_k. Each such subgraph can be considered as embedded with a number of patterns where each pattern is formed by combining an attribute value from each of e_i and v_i and $v_i^{'}$. Assume that the attribute values associated with e_i and v_i and $v_i^{'}$ are represented as $A^{e_i} = \{ e_{i,1}, ..., e_{i,j}, ..., e_{i,I_e} \}$, $e_{i,j} \in domain(A_{i,j}^{e^k})$, $j = 1, \ldots, I_e$, and $A^{v_i} = \{ v_{i,1}, ..., v_{i,j}, ..., v_{i,I_v} \}$ and $A^{v_i^{'}} = \{ v_{i,1}^{'}, ..., v_{i,j}^{'}, ..., v_{i,I_v}^{'} \}$, $v_{i,j}$, $v_{i,j}^{'} \in domain(A_{i,j}^{v^k})$, $j = 1, \ldots, I_v$, respectively. Let us first form the cross product $A^{v_i, e_i, v_i^{'}} = A^{v_i} \times A^{e_i} \times A^{v_i^{'}}$ so that each element in $A^{v_i, e_i, v_i^{'}}$ is a 3-tuple consisting of a particular combination of attribute values $(v_{i,j}, e_{i,j}, v_{i,j}^{'})$ of e_i and v_i and $v_i^{'}$. Each of these 3-tuple can be considered as a pattern. Such a pattern may or may not be

interesting. An interesting pattern has to be useful for characterization the PPI network of a species and for it to be discriminated against another

To determine if a pattern is interesting, we consider how frequently it appears in an MA graph G_k and compare it with how frequently it is expected to appear. If the difference is significant enough, it means that the pattern appears more or less frequently than expected and it can therefore be considered an interesting pattern in G_k. In other words, we are determining the probability of a graph G being part of the PPI of a particular species given that the pattern ($v_{i,j}, e_{i,j}, v'_{i,j}$) appears in G_k. This probability is compared against the probability that a subgraph S being the PPI of G_k. If the differences between them are not significant, it means that the pattern does not provide any information for one to decide if G should be part of G_k. Otherwise, it is interesting.

The first probability that we are interested in is the probability that a graph G, being part of the PPI of a particular species G_k given that the pattern appears in S. In other words, we are interested in the following probability:

$\Pr(G \subseteq G_k \mid \text{pattern is in S})$

$$= \frac{\text{Number of times pattern is in G and G is in } G_k}{\text{Number of times pattern appears in all } G_k} \tag{15}$$

And how much this probability differs from the following apriori probability

$$\Pr(G \subseteq G_k) = \frac{\text{Number of times pattern appear in } G_k}{\text{Number of times pattern in appear in all } G_k} \tag{16}$$

i.e., whether or not G is characterized by the pattern makes very little difference, then pattern should not be considered very interesting in determining if G should be classified into G_k. Otherwise, it can be very interesting.

To objectively determine if the two probabilities are different, we make use of a test statistic, d_{sp} which is defined [12].

$$d_{sp} = \frac{z_{sp}}{\sqrt{\gamma_{sp}}} \tag{17}$$

where z_{sp} is defined as:

$$z_{sp} = \frac{\Pr(\mathbf{G} \subseteq \mathbf{G}_k \mid \mathbf{G\ characterized\ by\ Pattern}) - n \Pr(\mathbf{G} \subseteq \mathbf{G}_k) \Pr(G\ \mathbf{characterized\ by\ pattern})}{\sqrt{n \Pr(\mathbf{G} \subseteq \mathbf{G}_k) \Pr(\mathbf{G}_k\ \mathbf{characterized\ by\ pattern})}} \tag{18}$$

and γ_{sp} is the maximum likelihood estimate of the variance of z_{sp} and is given by

$$\gamma_{sp} = (1 - \Pr(\mathbf{G} \subseteq \mathbf{G}_k))(1 - \Pr(\mathbf{G\ characterized\ by\ pattern})) \tag{19}$$

Based on [12], if $|d_{sp}| \geq 1.96$, we can conclude that the difference between $\Pr(\mathrm{G} \subseteq \mathrm{G}_k \mid \mathrm{G}$ characterized by pattern) is significantly different from $\Pr(\mathrm{G} \subseteq \mathrm{G}_k)$ and therefore the pattern is interesting and useful for classification. If $d_{sp} \geq +1.96$, it implies that the presence of the pattern in G provides evidence supporting G to be classified into G^k otherwise if $d_{sp} \leq -1.96$, it implies that the presence of the pattern provides negative evidence against G to be classified into G_k. In either case, pattern qualifies to be an *interesting* subgraph.

With the use of the adjusted residual analysis, *MISPAG-MA* screens each pattern in each graph $k = 1, \ldots, K$, to retain only those who are interesting. The set of interesting subgraph discovered for each of PPI network $\mathfrak{G}^{(1)}, \ldots, \mathfrak{G}^{(p)}, \ldots, \mathfrak{G}^{(P)}$ respectively is denoted as $\mathbf{S}^{'(p)}$, $p = 1, \ldots, P$. With the interesting first-order MA pattern, we can generate the second-order MA pattern by combining the interesting first-order patterns as follows:

The set of interesting patterns is discovered in each PPI network consists of one-edge subgraphs and these are therefore called first-order patterns. These first order patterns can be combined to form second order patterns that consists of two connected edges as follows.

Each interesting first-order pattern $\mathbf{S}^{'1}$ is made up of two vertices and an edge and an attribute value. These first order patterns can be combined to form second-order patterns if they can be connected through a common vertex. For example, given two interesting first-order patterns, $S'^{1}_{1} = (v_{i,j}, e_{i,j}, v'_{i,j})$ and $S'^{1}_{2} = (v_{i,j}, e_{i,j}, v'_{i,j})$, if $(\mathbf{a}_i^{v} \cup \mathbf{a}_i^{v'}) \cap (\mathbf{a}_j^{v} \cup \mathbf{a}_j^{v'}) \neq \phi$ that means S'^{1}_{1} and S'^{1}_{2} are sharing the same attribute value. A second-order MA pattern $\mathbf{S}^{'2}$ will be formed between $(v_{i,j}, e_{i,j}, v'_{i,j}, e_{i,j}, v'_{i,j})$ so that each second-order pattern contains three vertices and two connected edges. This expanding process is performed iteratively until no interesting patterns are discovered.

These patterns can be screened using the above procedures to form the next level candidate pattern until no qualified candidate can be further discovered (i.e.

$|d_{sp}| \leq 1.96$) and no candidate pattern can be generated (i.e. all interesting candidate subgraph has no connected edge for expansion). The final set of qualified candidate subgraph represents the specific class of network as interesting subgraph.

At each level, all possible subgraphs are verified with the interestingness measure to determine if it appears more frequently in class p than class $q \in P$ and $p \neq q$, and the details of determining the interestingness will be given in the next section. If a candidate subgraph is qualified, it will be used to generate the next level candidate subgraph.

6.3 Measuring Interestingness as a Function of the Weight of Evidence

The discovered interesting subgraphs provide positive or negative evidence supporting or refuting the classification of a graph into a particular class. *MISPAG-MA* measures how interesting these subgraphs are with the use of an interestingness measure defined in terms of an information-theoretic weight-of-evidence measure.

The more interesting a frequent subgraph is for a class, the greater the difference is between the two probabilities of $\Pr(G_k \in \mathfrak{G}^{(p)} | G_k$ is characterized by $S_{k,s}^{(p)})$ and $\Pr(G_k \in \mathfrak{G}^{(p)})$. Hence, the interestingness measure is defined again as a function of these two probabilities. Specifically, the more interesting $S_{k,s}^{(p)}$ is, the greater is the ratio between $\Pr(G_k \in \mathfrak{G}^{(p)} | G_k$ is characterized by $S_{k,s}^{(p)})$ and $\Pr(G_k \in \mathfrak{G}^{(p)})$. This ratio

can be measured with a *mutual information* measure, I($G_k \in \mathfrak{C}^{(p)}$: G_k is characterized by $S_{k,s}^{(p)}$), between $G_k \in \mathfrak{C}^{(p)}$ and $\mathbf{G}_k$ is characterized by $S_{k,s}^{(p)}$ as follows:

$$I(G_k \in \mathfrak{C}^{(p)} : G_k \text{ is characterized by } S_{k,s}^{(p)}) = \log \frac{\Pr(G_k \in \mathfrak{C}^{(p)} \mid G_k \text{ is characterized by } S_{k,s}^{(p)})}{\Pr(G_k \in \mathfrak{C}^{(p)})} \quad (20)$$

Based on the mutual information measure, the weight of evidence provided by $S_{k,s}^{(p)}$ for or against the classification of G_k into $\mathfrak{C}^{(p)}$ can be defined as:

$$\begin{aligned} W^{(p)}(G_k \mid S_{k,s}^{(p)}) &= W(G_k \in \mathfrak{C}^{(p)} / G_k \notin \mathfrak{C}^{(p)} \mid G_k \text{ is characterized by } S_{k,s}^{(p)}) \\ &= I(G_k \in \mathfrak{C}^{(p)} : G_k \text{ is characterized by } S_{k,s}^{(p)}) - I(G_k \notin \mathfrak{C}^{(p)} : G_k \text{ is characterized by } S_{k,s}^{(p)}) \end{aligned} \quad (21)$$

$W^{(p)}(G_k \mid S_{k,s}^{(p)})$ can be interpreted as a measure of the difference in the gain in information when a graph G_k contains $S_{k,s}^{(p)}$ is classified into $\mathfrak{C}^{(p)}$ as opposed to other classes. $W^{(p)}(G_k \mid S_{k,s}^{(p)})$ is positive if $S_{k,s}^{(p)}$ provides positive evidence supporting the classification of G_k into $\mathfrak{C}^{(p)}$, otherwise it is negative.

6.4 Using the Total Interestingness Measure for *Class*ification

Given the interesting subgraphs, $S'^{(p)}_{k,1}$, ..., $S'^{(p)}_{k,s'}$, ..., $S'^{(p)}_{k,s'_{ks'}}$} where p = 1, …, P discovered for each corresponding p classes, $\mathfrak{C}^{(1)}$, …, $\mathfrak{C}^{(p)}$, …, $\mathfrak{C}^{(P)}$, an "unseen" graph, G_U, not originally in G, can be classified by matching it against the interesting subgraphs in each of $\mathbf{S}'^{(p)}$, p = 1, …, P.

For every interesting subgraph, $S'^{(p)}_{s'} \in \mathbf{S}'^{(p)}$ that G_U matches, there is some evidence, $W^{(p)}(G_U | S'^{(p)}_{s'})$ provided by it for or against the classification of G_U into $\mathfrak{C}^{(p)}$. Assuming that G_U matches with $m_p \leq m'_p$ interesting frequent subgraph $S'^{(p)}_1$, …, $S'^{(p)}_{s'}$, …, $S'^{(p)}_{m_p} \subseteq \mathbf{S}'^{(p)}$, *MISPAG-MA* then computes a total interestingness measure for G_U to be classified into $\mathfrak{C}^{(p)}$. This total interestingness measure is defined as the summation of the total weight-of-evidence provided by each individual interesting frequent subgraph $S'^{(p)}_{s'}$ for or against G_U to be classified into $\mathfrak{C}^{(p)}$ as follows.

$$\begin{aligned} W^{(p)}(G_U) &= W(G_U \in \mathfrak{C}^{(p)} / G_U \notin \mathfrak{C}^{(p)} \mid G \text{ is characterized by } S'^{(p)}_1,...,S'^{(p)}_{m_p}) \\ &= \sum_{s=1}^{m_p} W(G_U \in \mathfrak{C}^{(p)} / G_U \notin \mathfrak{C}^{(p)} \mid G \text{ is characterized by } S'^{(p)}_s) \end{aligned} \quad (22)$$

The value of $W^{(p)}(G_U \mid S'^{(p)}_{s'})$ increases with the number and strength of the matched subgraphs in $S'^{(p)}_1,...,S'^{(p)}_s,...,S'^{(p)}_{m_p}$ that provide positive evidence supporting G_U to be classified into $\mathfrak{C}^{(p)}$ whereas the value of $W^{(p)}(G_U \mid S'^{(p)}_{s'})$ decreases if some matched subgraphs provide negative evidence refuting the classification of G_U into $\mathfrak{C}^{(p)}$. The total interestingness measure for G_U to be classified into each of $\mathfrak{C}^{(1)}$, …, $\mathfrak{C}^{(p)}$, …, $\mathfrak{C}^{(P)}$ is determined and *MISPAG-MA* assigns G_U to the class which give the greatest total interestingness measure.

Given the interestingness score of each subgraph, we can classify an unknown graph shown in *Figure* 7 by computing the total weight of interestingness

measure for it to be classified into each class. Given that only $S_4^{(p)}$, $S_5^{(p)}$, $S_6^{(p)}$, $S_7^{(p)}$, $S_8^{(p)}$, and $S_9^{(p)}$ are interesting frequent subgraphs in Table 7. Using Equation (22),

$$W^{(1)}(G) = W(Class = 1 / Class \neq 1 \mid S_6^{(p)}, S_{10}^{(p)}, S_{15}^{(p)})$$

$$= W(Class = 1 / Class \neq 1 \mid S_6^{(p)})$$

$$= \log_2 \frac{1/10}{6/20}$$

$$= -1.585$$

Similarly, $W^{(2)}(G) = 2.322$ and $W^{(3)}(G) = -1.585$. As the value of $W^{(2)}(G)$ in *Class* 2 is the largest among three classes, we can conclude that the unknown sample belongs to *Class* 2. Besides, there is negative evidence against the test graph being classified in *Class* 1 and 3, so this sample is not likely to belong to *Class* 1 or 3.

Compared to algorithms that classify graphs by considering only frequent subgraphs, Instead of relying solely on the appearance of frequent subgraph during classification, *MISPAG-MA* takes into consideration only those which are useful and interesting only. These frequent subgraphs are unique and can have biological meaning. The other frequent graph mining algorithms can only handle single class of data, if there are two or more classes, the comparative effect of a subgraph across all classes are ignored. There is always a chance that two or more classes have the same frequent subgraph. With interestingness measure, we can distinguish interesting frequent subgraphs from uninteresting ones for multiple classes.

6.5 Experiments and Results

The PPI network dataset is collected from the DIP database [15] that stores and organizes the experimentally determined interactions between proteins. It has captured the detailed information of the molecular interactions in the database. DIP is one of the reliable sources of PPI data among the available databases. All the DIP data can be accessed online in both interactive and batch modes. In this paper, we choose eight sets of PPI data from the species-specific sets available on Jan 26, 2009, and one from high-throughput genome scale dataset. The properties of the datasets we used in our experiments are listed in *Table* 17.

Table 17. Properties of the experimental datasets

Dataset	Species	No. of proteins	No. of interactions
Gavin	S. cerevisiae	1361	3222
Celeg	C. elegans	2653	4043
Dmela	D. melanogaster	7504	22871
Ecoli	E. coli	1878	7001
Hpylo	H. pylori	713	1423
Hsapi	H. sapiens	1815	2171
Mmusc	M. musculus	728	632
Rnorv	R. norvegicus	251	198
Scere	S. cerevisiae	4971	17611

Gavin contains the protein-protein interactions of S. cerevisiae identified by TAP purification of protein complexes followed by mass-spectrometric identification

of individual components, and the remaining eight datasets are the species specific subsets of DIP that contains all the interactions from the corresponding species as shown in *Table* 17. In each DIP file, there is a set of attributes characterized each interaction. They are: ID interactor A, ID interactor B, Alt. ID interactor A, Alt. ID interactor B, Alias(es) interactor A, Alias(es) interactor B, Interaction detection method(s), Publication 1st author(s), Publication Identifier(s), Taxid interactor A, Taxid interactor B, Interaction type(s), Source database(s), Interaction identifier(s), and Confidence value(s). In our paper, we only use three fields, ID interactor A, ID interactor B and Interaction type(s), which indicate the associations between protein A and protein B. With the ID of the proteins, we can find their corresponding functions from the UniProt Knowledgebase (UniProtKB) [18]. UniProtKB is the collection of functional information on proteins with rich annotations including biological ontologies, classifications, cross-references, evidence attribution of experimental and computational data, and amino acid sequence, etc.

6.5.1 Matching functions in the UniProtKB

The function of each protein in the PPI network is available under the Gene Ontology (GO) in the UniProtKB. The GO annotation includes the information of biological process, cellular component and molecular function. Usually, one protein has more than one function, for example, a protein with DIP ID "DIP- 5576N" refers to the Myosin light chain 1 protein in the species of Saccharomyces cerevisiae with five molecular functions: calcium ion binding, identical protein binding, motor activity, myosin II heavy chain binding and myosin V binding. The matching index

of protein ID and molecular functions is constructed to facilitate the function matching. We extracted 376,721 proteins with at least one GO annotation from the complete UniProtKB/Swiss-Prot data set. Each protein has a set of protein IDs including the DIP ID and accession number. However, we cannot use either DIP ID or accession number as the protein ID to match the functions. The reason is that only two percent of proteins have DIP ID in the UniProtKB/Swiss-Prot data set, and some protein in the DIP database has only DIP ID and no accession number. Hence, we use both DIP ID and accession number in the matching process. After the matching process, we found that there are lots of proteins with unknown function. The proportion of the proteins with unknown functions in each data set is shown in *Table* 18.

Table 18. The proportion of proteins with unknown functions

Dataset	No. of proteins	Proportion of unknown function
Celeg	2653	78%
Dmela	7504	87%
Ecoli	1878	14%
Hpylo	713	58%
Hsapi	1815	13%
Mmusc	728	12%
Rnorv	251	15%
Scere	4971	16%
Gavin	1361	7%

6.5.2 Discovery of Interesting Patterns

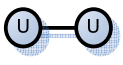
We applied three kinds of graph mining algorithms to discover sub-networks (subgraphs) from the above PPI network data sets. They are: Subdue that defines abstract subgraphs, *gSpan* that extracts frequent occurring subgraphs and MIMIC that discovers characterized and distinguished subgraphs. The representative sub-networks of each species discovered by these approaches are shown in *Table* 20 to 22 as below, and the vertex labels of these subgraphs are listed in *Table* 19.

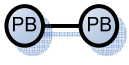
Table 19 List of protein vertex label

Code	Protein function	Code	Protein function
acs	1-aminocyclopropane-1-carboxylate_synthase	Mag	Magnesium ion binding
1D5	1-deoxy-D-xylulose-5-phosphate reductoisome	Me	Metal ion binding
35ex	3-5-exoribonuclease activity	mic	Microtubule binding
3dm	3-deoxy-manno-octulosonate cytidylyltransferase	msa	Microtubule-severing ATPase activity
ARA	Acetylcholine receptor activator activity	MoA	Monooxygenase actvity
act	Actin binding	MA	Motor activity
AMB	Actin monomer binding	Myo	Myosin binding
ata	Alanine transminase activity	Nic	Nickel ion binding
att	Arsentite transmembrane transporting ATPase	NA	Nucleic acid binding
ATP	ATP binding	Nu	Nucleoside binding
BLA	Beta-lactamase activity	NTA	Nucleoside triphosphatase activity
Ca	Calcium ion binding	NB	Nucleotide binding
CB	Calmodulin binding	Od	Odorant binding
CIA	Calmodulin inhibitor activity	PA	Peroxiredoxin activity
CA	Chemokine activity	PHA	Phosphopyruvate hydratase activity

Code	Protein function	Code	Protein function
CB	Chromatin binding	PLC	PLC activating metabotropic glutamate receptor
Co	Copper ion binding	PT	Positive transcription elongation factor activity
DNA	DNA binding	Pan	Protein anchor
DRO	DNA replication origin binding	PB	protein binding
DDP	DNA-directed DNA polymerase activity	PKA	Protein kinase activity
DRP	DNA-directed RNA polymerase activity	PPB	Pyridoxal phosphate binding
dic	Dynein intermediate chain binding	RA	Receptor activity
ECA	Electron carrier activity	RB	Receptor binding
EA	Endodeoxyribonuclease activity	RXR	Retinoid X receptor binding
en	Endoribonuclease activity	Rho	Rho guanyo-nucleotide exchange factor activity
ErbB	ErbB class receptor binding	RDR	Ribonucleoside-diphosphate reductase ctivity
FDA	Fomate dehydrogenase activity	Rib	Ribosome binding
GRT	General RNA polymerase II transcription factor	RNAII	RNA polymerase II transcription factor activity
GR	Glucocorticoid receptor binding	SMA	SMAD binding
gCd	Glutaryl-CoA dehydrogenase activity	SGR	Small GTPase regulator activity
GPA	Glycogen phosphorylase activity	sR	snoRNA_binding
GCR	G-protein coupled receptor activity	SB	Steroid binding
GFB	Growth factor binding	SH	Steroid hormone receptor activity
GTP	GTP binding	SCR	Structural constituent of ribosome
GAA	GTPase activator activity	SMA	Structural molecule activity
GA	GTPase activity	SDA	Succinate dehydraogenase activity
gne	Guanyl-nucleotide exchange factor activity	Su	Sugar binding
HeA	Helicase activity	TPR	TPR domain binding
Hem	Heme binding	TAA	Transcription activator activity
HB	Heparin binding	TCA	Transcription coactivator activity
HDA	Histone deacetylase activity	TCr	Transcription corepressor activity
HD	Histone deacetylase binding	TFA	Transcription factor activity
HLH	HLH domain binding	TFB	Transcription factor binding

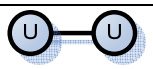
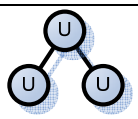
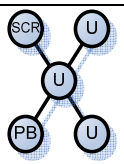
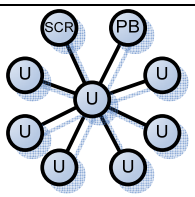
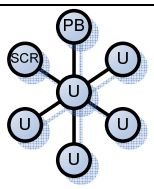
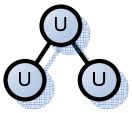
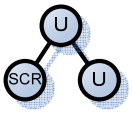
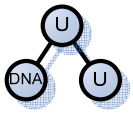
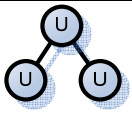
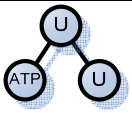
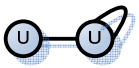
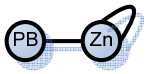
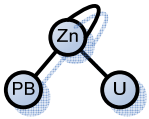
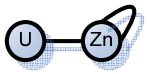
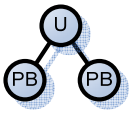
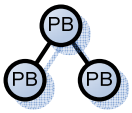
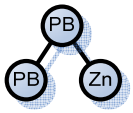
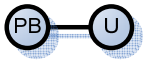
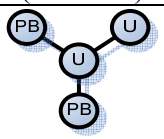
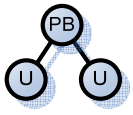
Code	Protein function	Code	Protein function
hsa	holo-[acyl-carrier-protein] synthase activity	TRA	Transcription repressor activity
HAB	Hyaluronic acid binding	TA	Transferase activity
hit	Hydrogen ion transporting APTase	TIF	Translation intiation factor activity
HIA	Hydrogen ion transporting ATPase activity	TRtk	Transmembrane receptor protein tyrosine kinase
HA	Hydrolase activity	ta	Transminase activity
IMP	IMP dehydrogenase activity	Ub	Ubiquitin binding
IGF	Insulin-like growth factor receptor binding	U	unknown
Ir	Iron ion binding	UA	Urease activity
LIF	Leukemia inhibitory factor receptor activity	Zn	Zinc ion binding
Lig	Ligase activity		

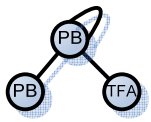
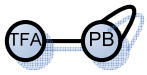
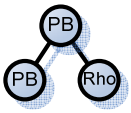
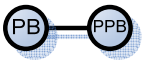
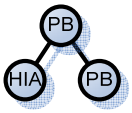
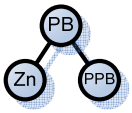
Subdue uses the MDL principle to discover subgraphs that best compress the original graph, and each discovered subgraph has a compression value to indicate its degree of compression. In the experiment, Subdue has discovered a large number of subgraphs from the PPI networks. The results are sorted by the compression value and users can specify the number subgraphs to show. The top five abstract subgraphs discovered by Subdue are listed in *Table* 11. Due to problem of incompleteness of protein function in some data sets, like the species of *Celeg* and *Dmela,* the discovered subgraphs mainly consist of the proteins with unknown function. However, these abstract subgraphs provide no information for characterizing a species class. Besides, some subgraphs like 

and 

are discovered in more than one species. It shows that different species have common sub-networks, but how these sub-networks can be compared across species is undefined.

Table 20 Abstract Subgraphs discovered by Subdue

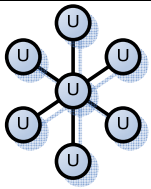
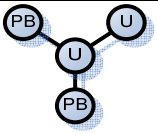
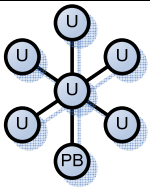
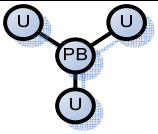
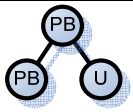
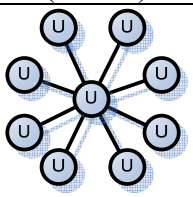
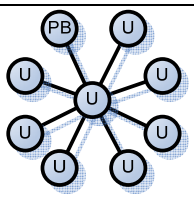
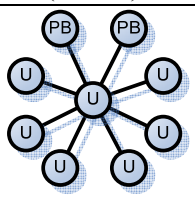
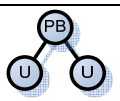
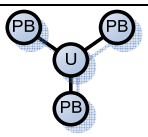
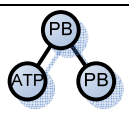
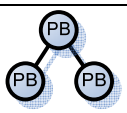
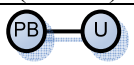
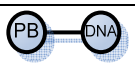
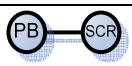
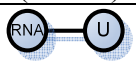
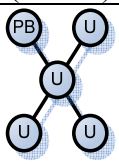
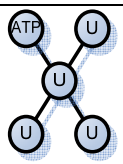
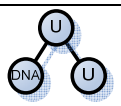
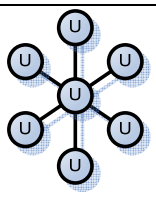
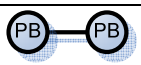
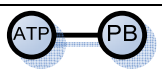
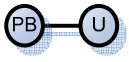
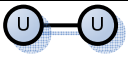
Species	**Top 5 abstract subgraphs** (compression value)				
Celeg	 (1.05526)	 (1.00963)	 (1.00577)	 (1.00471)	 (1.00458)
Dmela	 (1.05801)	 (1.00417)	 (1.00187)	 (1.00036)	 (1.00016)
Ecoli	 (1.02681)	 (1.01208)	 (1.00896)	 (1.00766)	 (1.00077)
Hpylo	 (1.02981)	 (1.02353)	 (1.01624)	 (1.0138)	 (1.01114)
Hsapi	 (1.03492)	 (1.03442)	 (1.03432)	 (1.03418)	 (1.1.03416)
Mmusc	 (1.03277)	 (1.00965)	 (1.005)	 (1.00234)	 (1.00226)
Rnorv	 (1.01729)	 (1.01633)	 (1.01612)	 (1.0157)	 (1.01437)

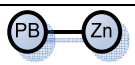
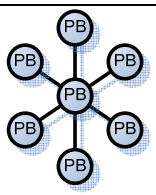
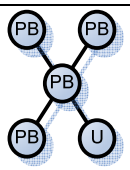
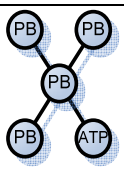
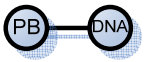
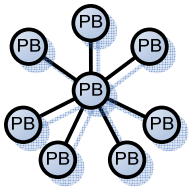
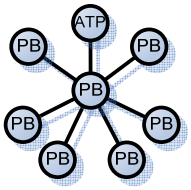
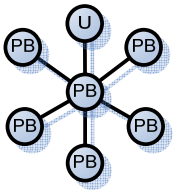
Species	**Top 5 abstract subgraphs** (compression value)				
Scere	 (1.00483)	 (1.00092)	 (1.00057)	 (1.00013)	 (1.00007)
Gavin	 (1.01455)	 (1.00101)	 (1.00057)	 (1.00043)	 (1.00003)

gSpan is an enhanced frequent subgraph mining algorithm that uses the DFS-based searching to discover frequent subgraphs. By setting the minimum support threshold to 3%, the maximal frequent subgraphs are identified in *Table* 21. *gSpan* calculates the occurring frequency of each frequent subgraph in a class, and it can potentially discover subgraphs that can characterize a class. The result shows that some protein functions are occurred frequently, like protein binding, ATP binding, DNA binding, structural constituent of ribosome, etc, and some protein interactions are occurring frequently in multiple species, for example, the protein interaction pair <protein binding - protein binding> is frequently occurred in the species of *Celeg*, *Ecoli*, *Hsapi*, *Mmusc*, *Rnorv, Scere* and *Gavin*. It helps to identify the common interactions that share across the species, but it is hard to characterize a class with these frequent subgraphs. Besides, suffering from the same problem of unknown protein function as mentioned above, *gSpan* discovers the frequent subgraphs with many vertices with unknown protein function. After filtering those

proteins with unknown function, the remaining subgraphs are only the fragments that contain a number of protein interactions.

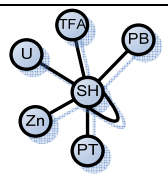
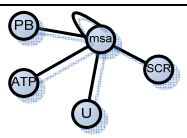
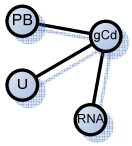
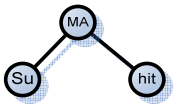
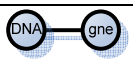
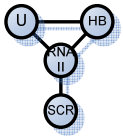
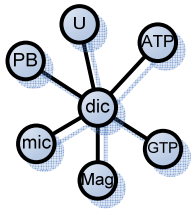
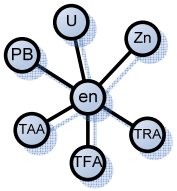
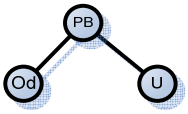
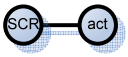
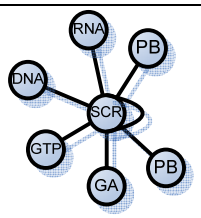
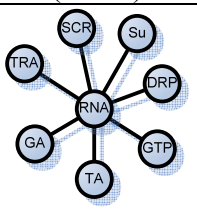
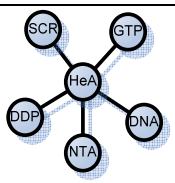
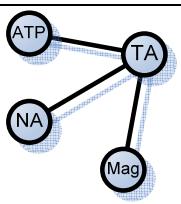
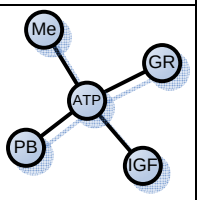
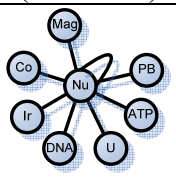
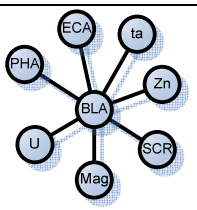
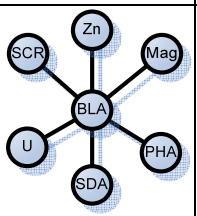
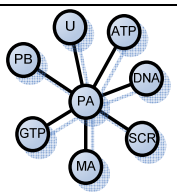
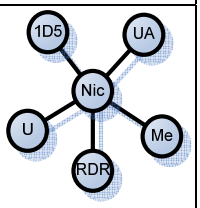
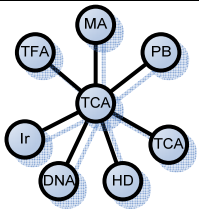
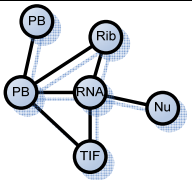
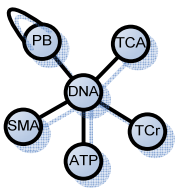
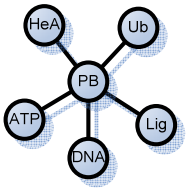
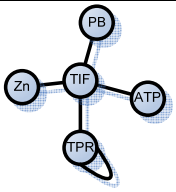
Table 21 Maximal Frequent Subgraphs discovered by gSpan (min. support = 3%)

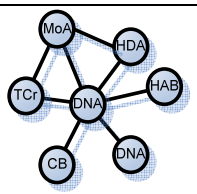
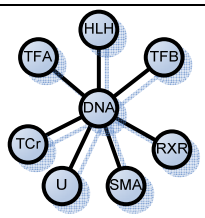
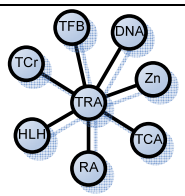
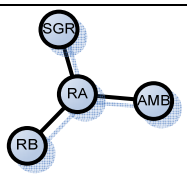
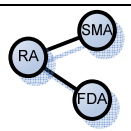
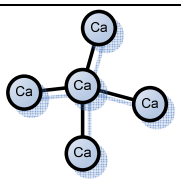
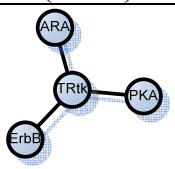
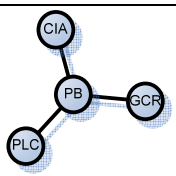
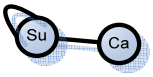
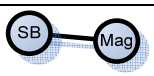
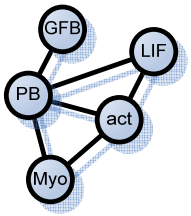
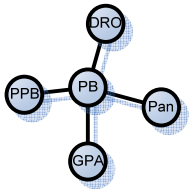
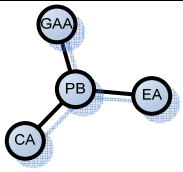
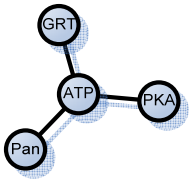
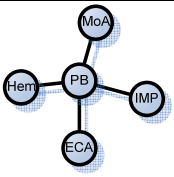
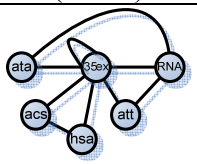
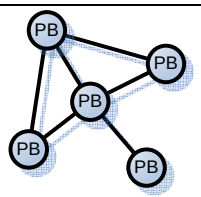
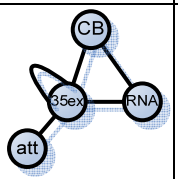
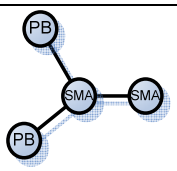
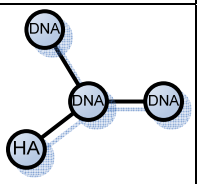
Species	**Top 5 maximal frequent subgraphs** (frequency)				
Celeg	 (5.50%)	 (5.31%)	 (5.19%)	 (5.19%)	 (5.15%)
Dmela	 (15.19%)	 (12.78%)	 (8.45%)	 (5.45%)	 (5.004%)
Ecoli	 (5.93%)	 (5.66%)	 (7.54%)	 (6.46%)	 (7.34%)
Hpylo	 (6.27%)	 (5.83%)	 (5.69%)	 (5.10%)	 (5.10%)
Hsapi	 (18.15%)	 (6.14%)			
Mmusc	 (9.98%)				
Rnorv	 (9.32%)	 (4.97%)	 (4.967%)		

Species	**Top 5 maximal frequent subgraphs** (frequency)				
Scere	 (10.17%)	 (8.46%)	 (8.04%)	 (8.00%)	
Gavin	 (5.16%)	 (4.56%)	 (3.57%)	 (3.57%)	 (3.19%)

To overcome the problems mentioned above, *MISPAG-MA* is used to discover interesting subgraphs that can characterize a class with an interestingness measure. It helps to identify the protein interactions for both characterization and classification. The five representative interesting subgraphs are selected as shown *Table* 22. With *MISPAG-MA*, the discovered interesting subgraphs are occurred frequently in one class more than the other classes than expected. It helps to identify the sub-networks that best represent a class, and it is useful to classify unknown network by comparing the interestingness measure across the species.

Table 22 Interesting subgraphs discovered by MISPAG-MA

Species	**Representative interesting subgraphs** (*d* value)				
Celeg	 (107.80)	 (37.33)	 (17.19)	 (7.63)	 (2.64)
Dmela	 (34.88)	 (32.54)	 (28.35)	 (7.38)	 (4.83)
Ecoli	 (172.8183)	 (119.19)	 (37.49)	 (19.05)	 (12.365)
Hpylo	 (91.36)	 (90.07)	 (82.77)	 (79.05)	 (66.423)
Hsapi	 (58.47)	 (51.89)	 (37.12)	 (29.03)	 (27.06)

Species	**Representative interesting subgraphs** (*d* value)				
Mmusc	 (67.04)	 (65.15)	 (53.56)	 (20.53)	 (8.90)
Rnorv	 (82.4)	 (62.02)	 (41.42)	 (23.52)	 (7.89)
Scere	 (32.19)	 (30.68)	 (23.97)	 (20.59)	 (49.65)
Gavin	 (52.70)	 (52.22)	 (47.45)	 (12.96)	 (7.71)

Similar to the existing graph mining algorithms, *MISPAG-MA* will also discover subgraphs that appeared in multiple species. Unlike them, these interesting subgraphs are discovered with the degree of uniqueness across multiple species. If we want to classify an unknown network to one of the species, the *d* value will be calculated for each class, so that the unknown sub-network will be classified as the class with the largest *d* value. Let's consider an unknown sub-network in *Figure* 28.

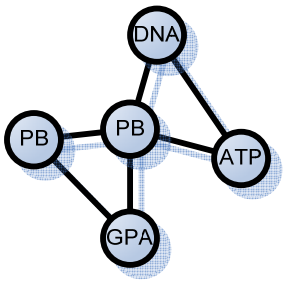
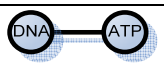
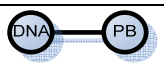
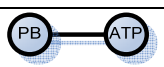
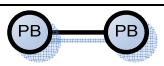
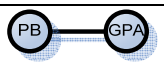


Figure 28 Unknown sub-network

The total weight-of-evidence of the unknown network is calculated by summing the individual interesting subgraphs as shown in *Table* 23. To simplify the illustration here, we use the first level subgraphs (edges) to calculate the total weight-of-evidence that reflects the degree of importance of the unknown network across species. After comparing the total weight-of-evidence across the species, we can classify the unknown network to be most likely to be appeared in *Scere* and most unlikely to be appeared in *Dmela*.

Table 23. The total weight-of-evidence of the first level subgraphs

$S_{k,s}^{(p)}$	$W(\mathbf{G}_k \in \mathfrak{T}^{(p)} / \mathbf{G}_k \notin \mathfrak{T}^{(p)} \mid \mathbf{G}_k \text{ is characterized by } S_{k,s}^{(p)})$								
	Celeg	Dmela	Ecoli	Gavin	Hpylo	Hsapi	Mmusc	Rnorv	Scere
	-3.53	-12.26	8.77	6.47	3.70	-4.71	-2.83	-1.29	3.35
	-4.06	-15.99	-4.68	9.96	-4.65	-5.58	-2.86	-1.70	17.53
	-1.94	-3.92	0.17	-2.86	-1.32	17.80	0.77	6.50	-5.69

	-5.73	-30.34	-21.65	8.70	-10.56	-15.22	-7.91	-4.90	51.89
	-0.75	-1.03	-1.67	-1.11	-0.51	-1.15	-0.55	-0.27	3.96
$\sum_{s=1}^{m_p} W^{(p)}(Gk \mid S_{k,s}^{(p)})$	**-16.01**	**-63.53**	**-19.05**	**21.15**	**-13.33**	**-8.86**	**-13.39**	**-1.66**	**71.04**

6.5.3 Performance Analysis

We tested all datasets using the two graph mining algorithms of Subdue and *gSpan* and then compare their results with our proposed graph mining algorithm MIMIC. *Table* 24 shows the performance of each algorithm on the different datasets.

From these results we can see that the classification performance of *gSpan* is better than Subdue by 8%. Their average classification accuracies are 0.301 and 0.326 respectively. With MIMIC, the overall performances have been improved by 120% and 100% respectively.

Table 24. Classification performance for FSG, gSpan and MISMOC

Dataset	***Class*ification Accuracy**		
	Subdue	***gSpan***	**MIMIC**
Gavin2002a	0.376	0.463	0.907
Celeg	0.181	0.188	0.379
Dmela	0.172	0.191	0.353

Ecoli	0.359	0.387	0.832
Hpylo	0.298	0.302	0.576
Hsapi	0.339	0.406	0.863
Mmusc	0.340	0.335	0.737
Rnorv	0.278	0.254	0.532
Scere	0.368	0.409	0.875
Average	**0.301**	**0.326**	**0.673**

The discovered interesting subgraphs discovered by *MISPAG-MA* are useful to classify an unknown network with the total weight-of-evidence based on the interestingness measure. The reasons are mentioned previously that the subgraphs discovered by many graph mining algorithms may appear frequently in a class but they may not uniquely represent a class. Subgraphs that may not appear very frequently can play an important role in discriminating one class from another. With MIMIC, the relative frequency of each subgraph is considered and how useful they are for classification are determined with a measure. The measure is then used when a graph is classified. This makes MIMIC more effective a graph classification algorithm.

6.6 Summary

This chapter addresses the problem of PPI prediction by modeling it as graph classification with the interesting subgraphs as input feature vectors for classification. As the previous subgraph mining algorithms have no classifier identified, we use a

well-known classifier *SVM* to perform the task of graph classification for *Subdue* and *gSpan*. For *MISPAG-MA*, we developed an interestingness measure to characterize each subgraphs for classification. By comparing the values of total weight-of-evidence in different species, we can correctly classify a sub-network to its corresponding species. We have shown that our algorithm can achieve a better AUC value in the ROC analysis over the other two algorithms. We also gave an example of determining an unknown network by our algorithm and discussed how the value of weight-of-evidence can be calculated for a given network. The results show that frequent subgraphs are unable to characterize a class as they may be commonly appeared in other species, while *MISPAG-MA* can define a set of interesting subgraphs by considering the relative frequency of each subgraph across multiple species, and specifically extract features that can be used to distinguish network of different species.

Chapter 7
Discovery of Class-Specific Patterns from Molecular Data

The elements that life is primarily made up of atoms: Carbon (C), Hydrogen (H), Oxygen (O), Nitrogen (N), and Phosphorus (P), etc. Atoms form molecules by various interactions and bonding: covalent bonds, ionic bonds, hydrogen bonds, hydrophobic interactions and van der Waals forces. Different atoms can form different numbers of covalent bonds, for example, hydrogen can form only one covalent bond, oxygen forms two, nitrogen forms three, and carbon forms four, etc. These structural patterns are pre-defined in the chemistry, but some patterns that are specific to the chemical molecules are not well defined.

The structure of the molecules stores the information that defines the functions of their physical and chemical properties. To identify a good ligand for a protein or DNA surface, one has to study the structure and function of the macromolecule. The structural information is essential for the drug discovery that makes use of the concepts of chemical similarity. Chemical similarity is measured by identifying distances between atoms on a receptor and a ligand. The chemical properties of the interacting atoms or group of atoms (functional groups) have a great influence on the reactivity of a ligand.

Several subgraph mining algorithms [9, 39, 23, 11, 40, 41] focus on the identification of the structural commonalities can be applied on mining the structure of ligands. One of the representative methods is *FSG* [40] that takes a set of graphs as input and a minimum support σ to find all connected subgraphs that occur in at least σ% of the graphs. *FSG* adopts a level-by-level candidate generation strategy that similar to Apriori algorithm. It has been applied on the biological domain for chemical carcinogenesis analysis [39] and [44] to find subgraphs typical to carcinogen of organic chlorides, mutagenesis data analysis [11] to identify subgraphs with higher or lower mutagenesis activity, and discovery of the anticancer therapeutics from the chemical compounds [44] available from the Developmental Therapeutics Program (DTP) at National Cancer Institute. Another subgraph mining algorithm based on depth-first search approach called *gSpan* can further speed up the costly candidate generation process of *FSG*. *gSpan* builds a lexicographic order among the graphs and maps each graph to a unique minimum DFS code as its canonical label. Based on this lexicographic order, *gSpan* adopts the depth-first search strategy to mine frequent connected subgraphs efficiently. It has been applied on a set of chemical compound data that contains 340 chemical compounds. The details of *FSG* and *gSpan* are described previously in Section 3.1.

Later, an optimized *gSpan* is proposed by Jahn et al. [72]. This version of *gSpan* presents two optimizations particularly tailored for databases of molecular graphs. The first optimization reduces the number of subgraph isomorphisms that need to be accessed for proper support computation in considering the symmetries inherent in many chemical molecules, and the second speeds up subgraph

isomorphism tests by making use of the non-uniform frequency distribution of atom and bond types. The optimizations are part of a reimplementation of the original *gSpan* algorithm and are shown to significantly increase the performance on two chemical datasets.

7.1 Data Description

The drug data from the KEGG ligand database currently contains 14,810 chemical compounds, 6,512 drugs, 10,959 glycans, 7,167 reactions, 7,187 reactant pairs. For our purposes, we selected three classes of drug data set, 1) Benzodiazepins, 2) Phenothiazines, and 3) Antivirals. All of these are collected from the ligand database, which contains chemical structures of all approved drugs in Japan and the U.S., together with additional information such as therapeutic categories and target molecules.

Each drug is represented in MOL format [23], which provides information about atoms, bonds, connectivity and the coordinates of a molecule for use in chemical structure comparisons. *Figure* 29 shows a sample MOL file. Line 1 shows the number of atoms and bonds (i.e. 9 atoms and 8 bonds). From line 2 to 10, the atom information like the atom type, coordinates are shown. From line 11 to 18, the bonding information like the atom IDs and bonding type are shown, for example, “2 3 1” in line 11 refers to a single bond between atom 2 and 3.

A chemical compound is a collection of atoms connected by covalent bonds. The atoms and bonds can be represented using a labeled graph in which all atoms are

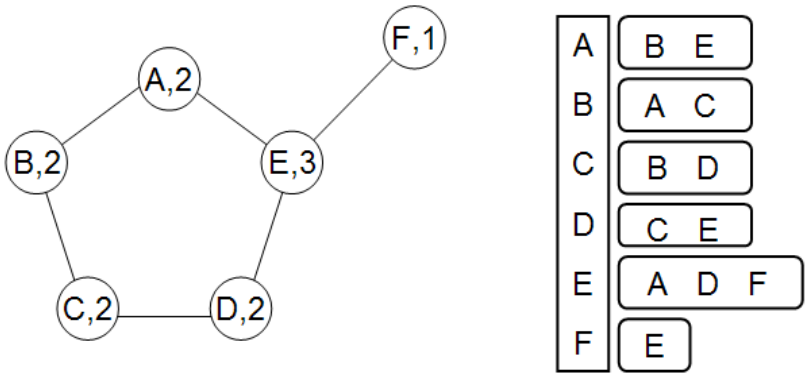
represented by attributed vertices and all bonds are represented by attributed edges. The same atoms in chemical compounds are distinguished by different labels as they represent different physiochemical properties in accordance with their spatial and chemical situations. We converted the data from MOL file format into graph transactions. Each atom in the compound is represented by a vertex, and each bond is represented by an edge.

```
9 8  0  0  0  0  0  0  0  0999 V2000
   12.6020  -27.3055    0.0000 C   0  0  0  0  0  0  0  0  0  0  0
   12.6020  -28.7058    0.0000 C   0  0  0  0  0  0  0  0  0  0  0
   13.8146  -29.4060    0.0000 C   0  0  0  0  0  0  0  0  0  0  0
   15.0274  -28.7058    0.0000 C   0  0  0  0  0  0  0  0  0  0  0
   15.0274  -27.3055    0.0000 C   0  0  0  0  0  0  0  0  0  0  0
   13.8146  -26.6053    0.0000 C   0  0  0  0  0  0  0  0  0  0  0
   16.2588  -26.5943    0.0000 C   0  0  0  0  0  0  0  0  0  0  0
   17.4643  -27.2902    0.0000 O   0  0  0  0  0  0  0  0  0  0  0
   16.2585  -25.2051    0.0000 O   0  0  0  0  0  0  0  0  0  0  0
  2  3  1  0     0  0
  1  4  2  0     0  0
  3  4  2  0     0  0
  4  5  1  0     0  0
  5  6  2  0     0  0
  1  6  1  0     0  0
  5  7  1  0     0  0
  7  9  2  0     0  0
M  END
```

Figure 29 MOL file format

7.2 Hierarchical Graph Representation

Each compound is represented as an attributed graph and then transformed into a set of hierarchical graphs. . To build these hierarchical graphs, we group components of the attributed graph into different levels according to their attributed structural relations. Hierarchical graph was introduced in Section 3.3 which represents hierarchical structural patterns that have multiple levels of complexity. It provides a means to group components (subgraphs) of the attributed graph in different levels according to whatever relation has induced the attributed vertex and edge sets. Highly connected graphs are always composed of multiple levels of concepts that can exist independently in form of subgraphs. *Figure* 31 shows the molecular structure of a drug molecule, Benzonate (Entry ID: D00242) from the KEGG database. It is first represented as an adjacency list in *Figure* 30, and then the five components, two cycles, one star and two linkages, are extracted as shown in *Figure* 32. The five components are then organized as vertices in a level 1 hierarchical graph as shown in *Figure* 33 that describes the component type (C: Cycle, S: Star and L: Linkage) and the total number of atoms (in subscript) of each vertex, and the edge is represented by the common vertex that shares between two adjacent components.



(a) (b)

Figure 30 (a) An undirected graph with six vertices (A to F) represented uniquely as (label, degree).(b) Adjacency list representation

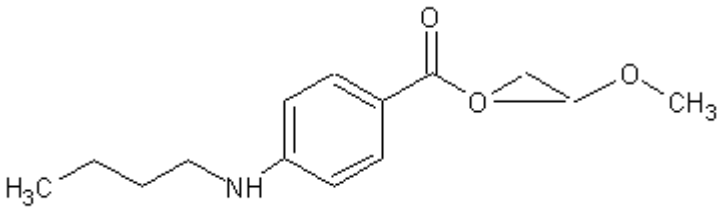


Figure 31 Molecular Structure of Benzonate

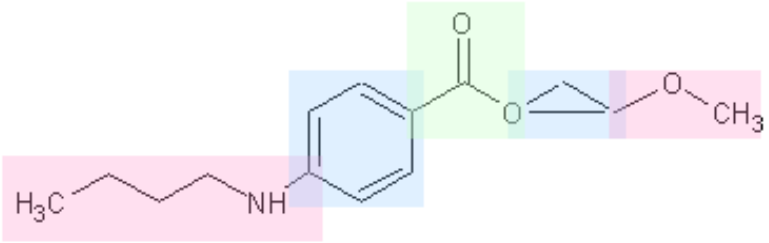


Figure 32 Component Extraction (cycle, star and linkage)

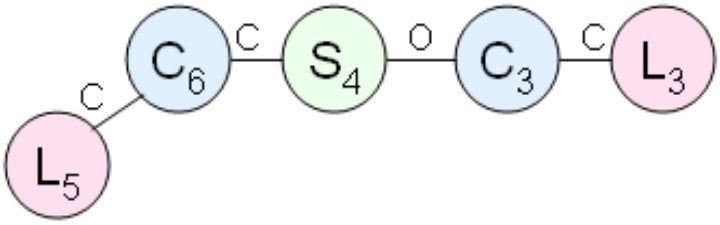


Figure 33 Hierarchical Graph of Benzonate

7.3 Experiments and Results

We use *FSG* and optimized *gSpan* to extract subgraphs occurring with a frequency above a given threshold (σ). If the classification accuracy is low, the value of σ is decreased in decrements of 10% until the maximum accuracy is achieved.

Figure 34 shows the classification accuracy of *FSG*, optimized *gSpan* and *MISPAG-MC* at σ values of 80%, 70% and 60%. At σ = 80%, the accuracy of *FSG* and *gSpan* are below 50% because the discovered frequent subgraphs are not useful in characterization. At σ = 70%, the classification accuracies are nearly the same for *FSG* and optimized *gSpan*, and *MISPAG-MC* still outperform the others. Then we further decrease the threshold by 10%. At σ = 60%, the improvements in accuracy associated with the *MISPAG* algorithms are very great, whereas *FSG* and optimized *gSpan* have only improved a little. This is because *MISPAG-MC* discovers only the discriminative patterns and robust to the noisy environment.. Although more subgraphs can be discovered at a lower support threshold by *FSG* and *gSpan*, they will capture more meaningless patterns at the same time. *MISPAG-MC* uses residual

analysis to extract the class-specific patterns and outperform *FSG* and optimized *gSpan* by over 55%.

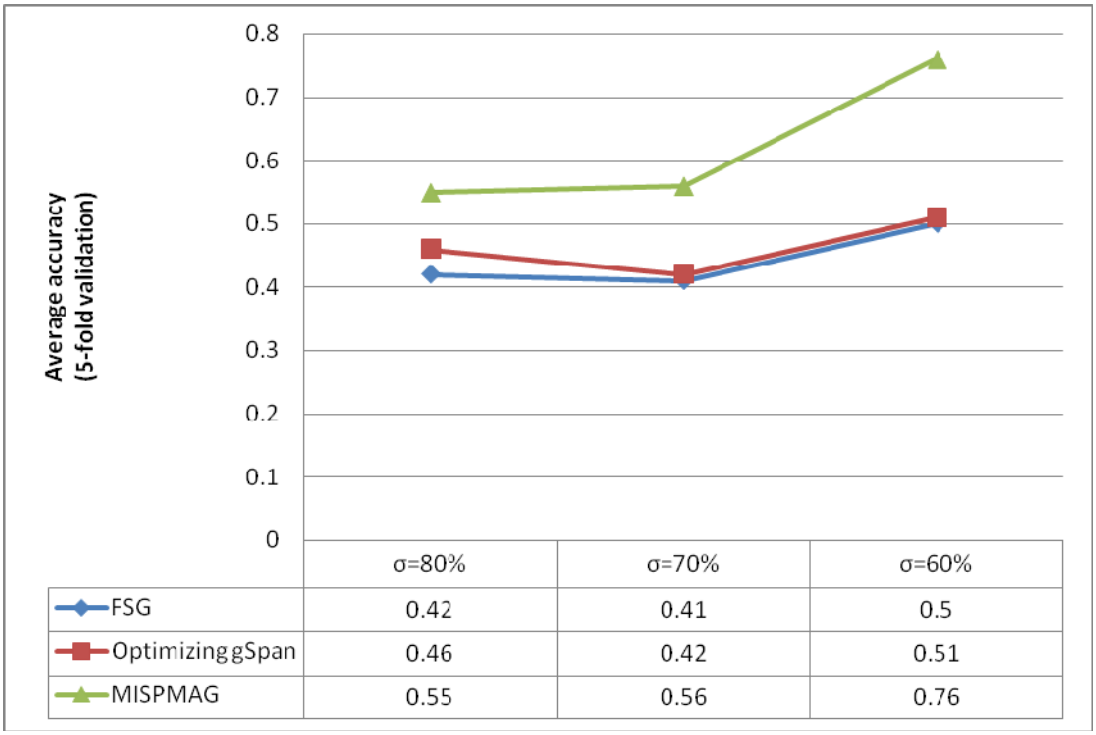
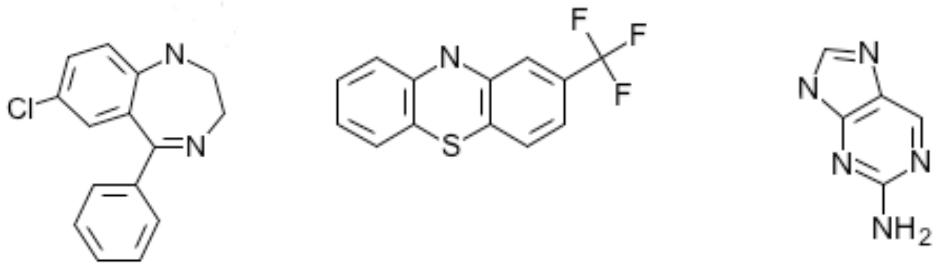


Figure 34 Classification accuracies of FSG, optimizing gSpan and MISPAG-MC

To speed up the classification process, we further introduced the hierarchical graph representation to group related atoms and bonds into a set of components. For example, the six carbon atoms in a benzene ring are represented as a cycle-6 component, and a level-1 hierarchical graph, MAG_1. In the first stage, the components are represented by the degree of connection. The extracted components are: Cycle-7 (heptagon), Cycle-6 (hexagon), Cycle-5 (pentagon), Star-4 (cross), Star-

3 (tripod), and the others are linkages. The interestingness measure of each component in each class is then calculated.

Table 25 shows interestingness measures for components of three classes of drug in a drug database. Some components in a class occur and some do not. For example, Cycle-7 is a positive class-specific component in class 1, and is not likely to occur in the other classes, especially not in *Class* 3, seeing as d_1 of Cycle-7 is greater than $+T$ and d_3 of Cycle-7 is less than $-T$. It is possible to form larger subgraphs by combining class-specific components with other adjacency components. An interesting pattern is selected from each class and they are shown in *Figure* 35.



(a) *Class* 1 (b) *Class* 2 (c) *Class* 3

Benzodiazepins Phenothiazines Antiviral

Figure 35 The interesting pattern in the three classes of drug: (a) Benzodiazepins (b) Phenothiazines and (c) Antiviral

After applying hierarchical graph representation, the average accuracy of *MISPAG-MC* with hierarchical graph representation is 81% at $\sigma = 60\%$. This shows

that the structure of the drug molecules is important to classification. By applying *MISPAG-MC* with hierarchical graph representation, the classification accuracy can be increased by filtering the noisy fragments, and at the same time speed up the subgraph discovery process.

Table 25 Interestingness measures of components in drug database

d		*Class* 1 Benzodiazepins	*Class* 2 Phenothiazines	*Class* 3 Antivirals
Cycle-5	C_2N_3	0.98	-3.34	2.20
Cycle-6	C_6	1.22	0.46	-1.68
Cycle-7	C_5N_2	3.96	-1.13	-2.96
Star-3	$C(C_3)$	-2.66	-0.92	3.59
Star-4	$C(CF_3)$	-3.81	2.71	1.24

7.4 Summary

This chapter demonstrates that our *MISPAG-MC* algorithm can successfully identify class-specific patterns from the molecule structure for classification. By identifying the highly discriminating patterns, our method can offer additional insight into the structural features that contribute to the chemical function of a compound. Many of the patterns denote various topologies such as carbon cycle and star, so the hierarchical graph provides a powerful and expressive representation for these chemical components. We have applied *FSG*, optimized *gSpan* and *MISPAG-MC* on

three classes of drug data. By comparing the classification result of the three algorithms, it indicated that the use of *MISPAG-MC* can discover discriminative patterns for classification.

Chapter 8
Discovering Protein Complexes with Biological and Structural Information in PPI Networks

Proteins usually interact with each other to accomplish vital functions, and these interaction relationships usually form massive PPI networks. PPI networks contain not only the information of individual interactions between protein-pairs, more biological meaningful pattern such as protein complexes are also included. Proteins usually perform functions in a group of two or more called protein complex [154, 155]. Such protein complexes play important roles in cells to perform many biological functions such as replication, transcription, and control of gene expression, etc. [112]. While a vast amount of protein interactions are detected by such high-throughput methods as mass spectrometry and yeast two-hybrid assays [64], the number of experimentally-determined protein complexes is still far from complete and keep growing [156]. As protein complexes are the key molecular entities to coordinate many biological functions and integrate multiple gene products to perform cellular functions, it is important to develop methods to accurately identify such protein complexes that are believed to be strongly evolutionary conserved [157]. Identifying protein complexes experimentally usually lead to high accuracy, however, it is both time-consuming and cost-expensive. Many computational algorithms have been developed to explore protein complexes from PPI networks. There are many large-scale PPI networks are derived for a variety of organisms, and such PPI

networks capture enormous interactions among proteins and can be represented in a graph structure where vertices and edges are used to represent proteins and the interactions among proteins respectively.

8.1 Existing Protein Complex Discovery Algorithms

The existing computational algorithms are usually based on such graph structure for clustering proteins into a set of protein complexes, including MCODE [158], CFinder [129], MCL [159, 160], DPClus [161], and IPCA [162].

MCODE [158] is a graph theoretic clustering algorithm that discovers protein complexes based on the connectivity between the proteins in a PPI network. Assuming that densely connected regions in the PPI network may represent protein complexes [163], it performs a local search algorithm that operates in three stages: vertex weighting, complex prediction, and an optional step of post-processing. It first weights all vertices (proteins) based on the local network density using the highest *k*-core of the vertex neighbourhood. Then it takes the highest weighted vertex as the seed vertex, and searches its neighbours recursively to include a new vertex with weight larger than a given threshold. Finally it filters all single core complexes and expands the remaining complexes by including neighbours with certain neighbourhood density. As highly weighted vertices may not be highly connected to each other, the vertices in the discovered pattern may not be densely connected. This violates the assumption of protein complexes are densely connected regions.

CFinder [129] is another graph clustering algorithm that targets on discovering a group of densely interconnected nodes, called modules, in the network graph. It uses the Clique Percolation Method [164] to locate all *k*-clique clusters that corresponding to the fully connected subgraphs of *k* vertices. A larger value of *k* indicates that the density of a cluster is higher, however, real protein complexes are not always connected in form of a clique. To relax this constraint, an adjacency *k*-clique cluster is defined as two *k*-clique clusters that share *k*-1 nodes with each other, and the module that corresponding to a protein complex is defined as the union of *k*-clique clusters that can be reached from each other through a series of adjacency *k*-clique clusters. However, problems exist at both large and small values of *k* in this algorithm. A large value of *k* will miss many complexes that could not form a *k*-clique clusters as the interactions in PPI networks are not fully detected [14, 15, 16]; and a small value of *k* will include many false complexes as PPI networks contain many such small clusters and many noisy clusters will be merged.

MCL [159, 160] is a graph clustering algorithm that finds clusters in a graph based on Markov matrices to perform the concept of random walk [165]. The random walk that visits a dense region in a graph will likely not leave the region until many of its vertices have been visited. By simulating the random walk by means of the expansion and inflation operators, an arbitrarily selected vertex is able to search for new neighbours, and further promoted to include new vertices or demoted to disconnect with the sparsely connected regions, respectively. With MCL, the discovered subgraphs are not necessary to be cliques with high density if the

vertices within the subgraph have relatively fewer interactions with the vertices outside the subgraph.

DPClus [161] is a graph clustering algorithm that extracts densely connected regions from a graph as protein complexes. It first represents the input PPI network graph as an associated matrix, and set a minimum threshold of the density and cluster property for cluster generation. In the graph, each edge is assigned with an edge weight that indicates the number of common neighbours between two connected vertices, and each vertex is assigned with a vertex weight that indicates the sum of weights of the edges connected to the vertex. A cluster is generated from the highest weight vertex as a seed node, and it is expanded by adding one vertex at a time from its neighbours with the priority of higher vertex weight and the connecting degree. If the resulting cluster meets the thresholds of density and cluster property, the vertex will be added to the cluster. Once a cluster is generated, new clusters are formed in the remaining graph iteratively until no edge is left.

Later, IPCA [162] is proposed to modify DPClus algorithm based on a new topological structure to identify protein complexes in PPI networks. IPCA follows the seed vertex selection approach of DPClus but modifies the rules of vertex weighting and cluster expansion. As many known complexes are very small in size (i.e. short vertex distance), the attention here is shifted to discover subgraphs with a shorter vertex distance. The vertex distance is limited by a maximum threshold of vertex distance to guarantee that the discovered subgraph will not grow to a very large size. The weights of the vertices are calculated only once based on the original

network instead of updating iteratively as they believe that the original network contains the dense structure of clusters, and re-computation will lead to a lost of biological information. The cluster is expanded by adding neighbouring vertices recursively according to a threshold of interaction probability that replaces the density and cluster property of DPClus.

These clustering algorithms for identifying protein complex in PPI networks are evaluated in different studies [162, 166]. These algorithms have a common characteristic that they all target on identifying densely connected regions from a graph based on such topological information as connecting degree and density. Experiments are performed on the large-scale PPI network of *Saccharomyces cerevisiae* and the predicted patterns are verified by the known protein complexes. In the two studies, MCL and IPCA are found to be the best algorithms in identifying protein complexes from the PPI network, respectively.

While protein complexes are biologically meaningful, graph clustering algorithms that are solely based on topological information to identify protein complexes may not be effective. All these algorithms assume that proteins in the same complex are densely connected, so they only target on such regions to identify protein complexes. However, the matching rates of these algorithms against the known protein complex data are not high as reported, and they usually generate many false positive results.

In this chapter, we applied *MISPAG-PC* to develop a novel graph clustering algorithm to identify protein complexes from PPI networks with a top-down

approach. Unlike the existing graph clustering algorithms that solely target on finding densely connected regions in PPI networks, we include such biological information as molecular function of proteins in the clustering process. The key idea here is to capture biological relationships between proteins, and define a significant score for each interaction for identifying interesting subgraphs as protein complexes. We believe that the proteins in a complex are not only structurally inter-connected, but also biologically related. We first represent the PPI network as a network graph with each vertex representing a protein and each edge representing the interaction between two proteins. Given such a PPI network graph, each vertex is labeled with the molecular function in Gene Ontology [147] that its corresponding protein performs. By means of an interestingness measure defines in terms of the observed and expected probabilities of occurrences, we can determine the interaction between two neighbouring proteins is interesting or not, and each vertex in the network graph is therefore connected to a group of vertices with different significant scores. The clustering process is divided into two steps: local filtering, and global extraction. In the local filtering, the maximum significant score is first obtained for each vertex, and the edges with significant score lower than the maximum score by a certain threshold will be filtered. After filtering those locally uninteresting edges, the remaining graph will undergo the global extraction to identify interesting subgraphs that are connected by the interesting edges.

We compare our proposed *MISPAG-PC* algorithm with five existing graph clustering algorithms: MCL, IPCA, DPClus, MCODE, and CFinder to identify protein complexes from three sets of PPI networks. The identified patterns are then

matched with the known protein complexes of MIPS [14] to obtain their recall rates at different overlapping degrees. A higher recall rate at higher overlapping degree corresponds to more accuracy result. From the experimental results, we found that the recall rate of *MISPAG-PC* is much higher than the other algorithms especially at higher overlapping degree. It proves that using topology information is not the only and best way to identify protein complexes, the biological and structural information of proteins are both important to be considered for protein complex identification.

8.2 PPI Network Representation and Annotation

The raw data of biological network is a labeled graph that treats each protein as a unique entity and ignores any other useful biological information. Since the functions of the proteins are already annotated in databases such as Gene Ontology (GO) [8], we can model each network as biological annotated template for further analysis. Biological annotation of biological networks helps to define, understand and compare essential cellular actions in different organisms that involving similar functional units. It also helps to determine the functional roles of the other un-annotated units. To match these functions to the proteins, a matching index of identifier and molecular functions is constructed to facilitate the function matching. However, a biological unit may represent in different ID in different databases [14], [15], [68], [69], [70], for example, P53141 in IntAct, 5576N in DIP, YGL106W in Ensembl and KEGG. For those biological network databases that are not identified by the UniProtKB/Swiss-Prot ID, mappings are developed between the databases in

order to locate the corresponding functions in GO. Besides, a biomolecule is usually annotated with multiple GO terms in the taxonomy of GO.

The GO project is initiated since 1998 to construct and use ontologies to facilitate a standardized representation of gene and gene products such as proteins in a wide variety of species. It can be used to annotate genes and proteins with the known biological attributes such as molecular function, biological process and cellular component. It should be noted that many genes and proteins may perform one or more molecular functions, involve in one or more biological processes, and locate in one or more cellular components, for example, UniProtKB/Swiss-Prot P53141 refers to the protein Myosin light chain 1 in the species of *Saccharomyces cerevisiae* with five molecular functions (calcium ion binding, identical protein binding, motor activity, myosin II heavy chain binding, and myosin V binding), three biological processes (cytokinesis, protein localization, and vesicle targeting) and four cellular components (cellular bud neck, cellular bud tip, myosin complex, and vesicle).

Many graph mining algorithms for identifying protein complexes such as MCODE [158], CFinder [129], and MCL [159, 160] are designed for detecting densely connected regions in PPI networks, and they ignored the detailed information of each protein. However a protein contains lots of biological information that are represented by the GO terms. The distribution of the GO attributes in the Gene Ontology database is shown in *Figure* 36.

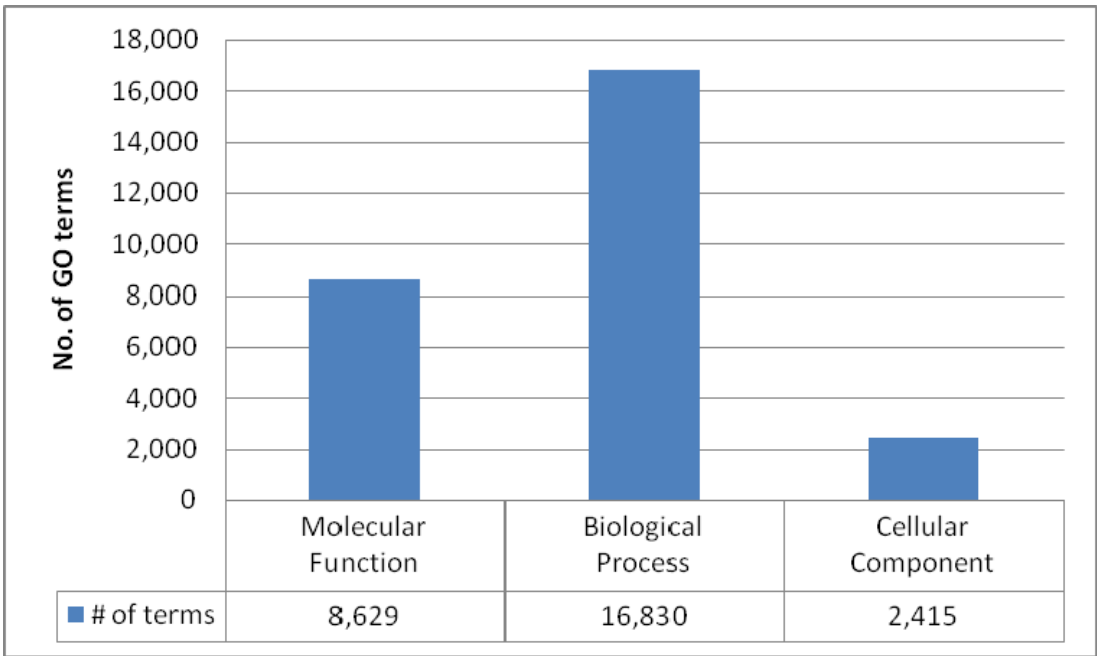


Figure 36 The distribution of the key attributes in Gene Ontology

Given a network graph G (**V**, **E**), traditional graph mining algorithms for identifying protein complexes represent a graph with a list of edges **E**, and all vertices in **V** are simply represented by its identity number. They treat the network graph as a simple undirected graph with no biological information. However, in the case of many real data such as biological network, the biomolecules (vertices) and interactions (edges) are usually defined by multiple biological attributes with multiple values. We should not ignore this information when discovering the biologically meaningful protein complexes.

8.3 Local Filtering of Uninteresting Interactions

Once the network graph labeled with the corresponding attribute values of its constituent proteins, is constructed, *MISPAG-PC* proceeds to discover interesting patterns in it. To do so, *MISPAG-PC* examines each vertex in turn to determine if its interactions with each of its neighbouring vertices are interesting. If so, the protein pairs forms part of a larger interesting subgraph. Otherwise, they are not considered for further processing. The process is carried out by three steps: interestingness measure, local filtering mechanism, and interesting subgraph formation.

8.3.1 Interestingness Measure

To determine if a vertex and any of its neighbouring vertices is interesting, an objective interestingness measure is used and it is defined previously in Chapter 4. Assume the protein that $v \in \mathbf{V}$ represents interacts with n other proteins, then v can be considered as connected to n other vertices, v_1, v_2, ..., v_n. To determine if an interaction between v and $v' \in \{v_1, v_2, ..., v_n\}$ is interesting, we first need to calculate the interestingness score between the attribute values of v and v'. The set of attributes associated with a vertex v can be represented as $\mathbf{A}^v = \{ A_1^v, ..., A_j^v, ..., A_{I_v}^v \}$ and each attribute $A_j^v, j = 1, \ldots, I_v$, can take on values from $domain(A_j^v) = \{ a_{j,1}^v, ..., a_{j,l}^v, ..., a_{j,J_v}^v \}$. The associations between two vertices v and v' is therefore constructed by the cross product $\mathbf{A}^v \times \mathbf{A}^{v'}$ so that each element is a 2-tuple pattern that defined similarly in Section 6.2. To determine if a pattern is interesting, we

consider how frequently it appears in the PPI network graph and compare it with how frequently it is expected to appear by the adjusted residual analysis. If the difference is significant enough, it means that the pattern appears more or less frequently than expected and it can therefore be considered an interesting pattern for constructing an interesting subgraph. To determine the difference between the conditional probability of the value of attribute *j* of *v* and the value of attribute *j'* of *v'*:

Pr(the attribute value of vertex on one side of an edge is a^{v}_{j,l_v} | the attribute value of vertex on the other side is $a^{v'}_{j',l_{v'}}$) = Pr($a^{v}_{j,l_v} \mid a^{v'}_{j',l_{v'}}$)

$$= \frac{\text{Total number of edges in G connecting vertices with } a^{v}_{j,l_v} \text{ and } a^{v'}_{j',l_{v'}}}{\text{Total number of edges in G that connects vertex with } a^{v'}_{j',l_{v'}} \text{ to other vertices}} \quad (23)$$

and the following apriori probability:

Pr(the attribute value of vertex on one side of an edge is a^{v}_{j,l_v}) = Pr(a^{v}_{j,l_v})

$$= \frac{\text{Total number of edges in G that connects } v\text{vertex with } a^{v}_{j,l_v} \text{ to the other vertices}}{\text{Total number of edges in G}} \quad (24)$$

is significantly different.

If the difference is significant, the interaction between the proteins corresponding to attribute values a^{v}_{j,l_v} and $a^{v'}_{j',l_{v'}}$ is considered interesting. This calculation is not limited to one attribute, it can also applied on vertex with multiple attributes and each attribute with multiple values.

Given an interaction between vertex v_a and vertex v_b, the sets of attributes associated with the two vertices are A^{v_a} and A^{v_b} respectively. Assume each vertex has two same attributes, i.e. $label(A_1^{v_a}) = label(A_1^{v_b})$, and $label(A_2^{v_a}) = label(A_2^{v_b})$, the attribute values of v_a is represented as $domain(A_j^{v_a}) = \{a_{j,1}^{v_a}, ..., a_{j,l_a}^{v_a}, ..., a_{j,J_{v_a}}^{v_a}\}$ and the attribute values of v_b is represented as $domain(A_j^{v_b}) = \{a_{j,1}^{v_b}, ..., a_{j,l_b}^{v_b}, ..., a_{j,J_{v_b}}^{v_b}\}$ where $j = 1$ and 2 that refer to the two attributes.

To allow interestingness to be compared, an interestingness measure called *adjusted residual* is defined in terms of the two probabilities in Equation (23) and (24) as follows:

$$d_{v_a v_b} = \max_{\substack{j=1,2 \\ l_a = 1,2 ..., J_{v_a} \\ l_b = 1,2..., J_{v_b}}} (\gamma[\Pr(a_{j,l_a}^{v_a} \mid a_{j,l_b}^{v_b}) - \Pr(a_{j,l_a}^{v_a})]) \tag{25}$$

where $\gamma = \dfrac{1}{\Pr(a_{j,l_a}^{v_a})\sqrt{\Pr(a_{j,l_a}^{v_a})(1 - \Pr(a_{j,l_a}^{v_a}))\Pr(a_{j,l_b}^{v_b})(1 - \Pr(a_{j,l_b}^{v_b}))}}$ and $d_{v_a v_b}$ has a standard normal distribution.

The interestingness score of an interaction between two vertices is obtained by selecting the maximum interestingness that refers to the most significant relationship between the two vertices. To facilitate the determination of interestingness of each pair of vertices, an interestingness scoring matrix of attribute values is constructed in advance. This scoring matrix serves as a rule-based to qualify each vertex pairs.

8.3.2 Local Filtering Mechanism

Each protein usually interacts with the other proteins in a PPI network. After obtaining the interestingness score for each interaction in the PPI network, we can filter the uninteresting edges as follows. For each vertex v_i in a PPI network graph G = (**V**, **E**), we first construct its neighbourhood graph $\mathcal{N}_i$ with v_i as a centroid and its connected neighbors $v_i' \in \{v_1, v_2, ..., v_{n'}\} - v_i$. For example, the neighbourhood graph $\mathcal{N}_1$ in *Figure* 37 contains a centroid v_1 and its connected neighbors v_2, v_3, v_4, v_5 and v_6.

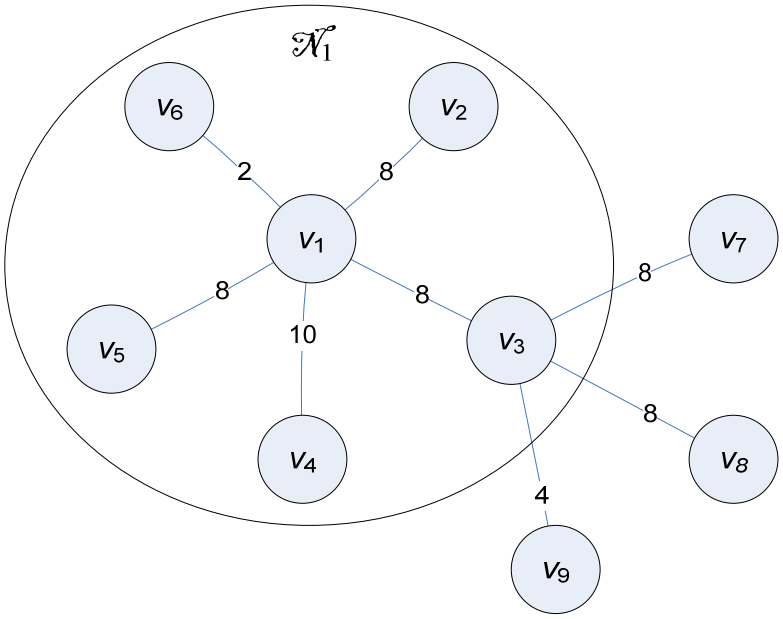


Figure 37 An example of neighbourhood graph

For each $\mathcal{N}_i$, i = 1, 2, …, n, with size greater than two (i.e. number of vertex is more than two), the score of the most significant interaction of the centroid vertex will be extracted as the base score b. By setting the local filtering threshold μ, we

can filter the uninteresting edges by defining the local interestingness threshold $I(v_i)$ for each vertex v_i is:

$$I(v_i) = b_i \times \mu \tag{26}$$

where $b_i = \max_{v_i'=\{v_1, v_2, \ldots, v_n'\}-v_i} (d_{v_i v_i'})$

If the interestingness score of v_i and v_i' is lower than $I(v_i)$, i.e. $d_{v_i v_i'} < I(v_i)$, then the interaction between v_i and v_i' is considered as uninteresting and filtered. Taking *Figure* 37 as example, assume μ is set to 0.8, the local interestingness threshold $I(v_1)$ of $\mathcal{N}_1$ is equal to 8 according to Equation (26), and the interaction between v_1 and v_6 with interestingness score equal to 2 will be filtered. This filtering scheme assumes that interesting interactions are more likely to connect with each other to form a biologically meaningful subgraph which may correspond to a protein complex. Eventually, the iterating filtering process will remove all uninteresting interactions, and only the interesting ones will be remained in the graph as separated subgraphs. This approach is a top-down approach to cluster a single network graph into different individuals. Comparing with the existing approaches that discover protein complexes by expanding from a single protein, our approach can work more efficiently during the extraction process.

8.4 Experiments and Results

To discover interesting patterns in PPI networks, we choose the high-throughput (also known as genome scale) data of *Sacchromyces cerevisiae* that made available in the Biological General Repository for Interaction Datasets (BioGRID) database [16] for our experiments. BioGRID is one of the public repositories that includes a virtually complete set of interactions reported to date from different sources. We collect two sets of PPI networks from Gavin et al. [112] and Krogan et al. [113], that are characterized by the mass spectrometry technique, are commonly used in many studies [109, 110, 111]. These sources have produced an enormous amount of PPI data of *S. cerevisiae* that allow us to perform a more complete analysis. After removing the self-connected and duplicate interactions, the remaining data sets that are used in our study are described in *Table* 26.

Table 26 Data description of the experimental data

Dataset	***Gavin***	***Krogan***
Number of proteins	1429	2663
Number of interactions	6527	7066
Number of proteins with unknown molecular function	101	336
Number of proteins with unknown biological process	44	259
Number of proteins with both unknown molecular function and biological process	18	135

The network graphs are constructed and labeled with the molecular functions and biological processes, as vertex attributes, performed by the proteins in the PPI networks. The cellular components are not selected as one of the attributes due to the high missing rate, and their values usually correspond to a protein complex name. In the labeling process, if both molecular functions and biological processes are missing, they are marked as unknown.

8.4.1 Evaluation Method

To evaluate the effectiveness of different algorithms for identifying protein complexes, a scoring scheme called overlapping score *o* [158] is used to determine how accurate a predicted pattern is matched with a known protein complex. The known protein complexes are retrieved from the MIPS: Comprehensive Yeast Genome Database [14], which is the most comprehensive public datasets of the protein complexes of *S. cerevisiae*. After removing the duplicate complexes and those contain only one protein, we obtained 1,049 known protein complexes from MIPS, and the largest one is a probably RNA metabolism protein complex (MIPS ID: 550.1.149) which contains 88 proteins. The overlapping score between the predicted pattern P and known protein complex C is defined as $o(P,C) = m / n_P * n_C$, where m is the number of common proteins shared by P and C, n_P is the number of proteins in P, and n_C is the number of proteins in C. The value of the overlapping score is ranging from 0 to 1. If o is equal to 0, it means no protein in C is found in P and vice versa (i.e. zero matching); if o is equal to 1, it means all proteins in P are found in C and vice versa (i.e. perfect matching). With this scoring scheme, we can measure the

biological significance of each predicted pattern. A predicted pattern and a known complex are considered as a match if their overlapping score is larger than or equal to a specific threshold.

8.4.2 Matching of known protein complexes

In order to compare the performance of different protein complex discovery algorithms, we use MCODE [158], CFinder [129], MCL [159, 160], DPClus [161], and IPCA [162] to predict protein complexes from the selected PPI networks. According to the previous comparative analysis of Brohee et al. [43] and Li et al. [162], the parameters of these algorithms are set as *Table* 27 to obtain the best matching results. We use a single attribute, molecular function, to label the proteins with multiple values, in our proposed algorithm *MISPAG-PCSA*. The parameter μ is set to 0.8 in the local filtering process.

Table 27. Parameter settings of different algorithms

Algorithms	**Parameters**
MCODE	*Node Score Cutoff* = 0.2 *Haircut* = 0 *Fluff* = 0
CFinder	$k = 3$
MCL	$I = 2$ *resource* = 4
DPClus	$CP = 0.5$ $D = 0.9$
IPCA	$T = 0.9$

	$S = 2$ $P = 2$

Each algorithm is based on the recommended settings discovers a list of predicted patterns. These patterns are then matched with the known protein complexes in MIPS as mentioned in Section 8.4.1. The performances of different algorithms are given in *Table* 28 and 29 that corresponding to the datasets of *Gavin* and *Krogan*, respectively.

In the dataset of *Gavin*, the algorithm MCL and our proposed *MISPAG-PCSA* outperform the other four algorithms with overlapping score from 0.6 to 1.0. Among the algorithms, IPCA predicts the largest number of patterns, and MCODE predicts the smallest number of patterns. However, MCODE gives a better result than IPCA in most cases; it shows that the number of false positive patterns of MCODE is much less than IPCA. At $o \geq 0.6$, the number of matched known protein complexes by MCL is the best, *MISPAG-PCSA* is the second, and CFinder is the third. The performance of MCL and *MISPAG-PCSA* are nearly the same, and they match 1.74 and 1.64 times more than CFinder. At a higher overlapping score, $o \geq 0.8$, the performance of *MISPAG-PCSA* is better than MCL by 1.04 times, and their results are still better than the other algorithms. At $o \geq 0.9$ and $o \geq 1.0$, *MISPAG-PCSA* obtains a better result than MCL by 1.34 and 1.35 times, respectively. With the same overlapping score, some algorithms such as IPCA and DPClus cannot predict any protein complex. The results show that *MISPAG-PCSA* outperforms all these five

existing algorithms on the performance of identifying protein complexes at $0.8 \leq o \leq 1.0$.

Table 28 Comparison of the number of predicted patterns generated by different algorithms that match known protein complexes on the dataset of Gavin

Algorithm	**Total no. of predicted patterns**	***o* ≥ 0.6**	***o* ≥ 0.7**	***o* ≥ 0.8**	***o* ≥ 0.9**	***o* = 1.0**
MCL	232	101 (1st)	72 (1st)	47 (2nd)	32 (2nd)	31 (2nd)
IPCA	818	50	8	4	0	0
CFinder	97	58 (3rd)	42 (3rd)	27 (3rd)	11	10
DPClus	285	45	15	4	0	0
MCODE	68	42	29	21	13 (3rd)	13 (3rd)
MISPAG-PCSA	223	95 (2nd)	64 (2nd)	49 (1st)	43 (1st)	42 (1st)

In the dataset of *Krogan*, our proposed *MISPAG-PCSA* algorithm outperforms all the other five algorithms with overlapping score from 0.6 to 1.0. Similar to the dataset of *Gavin*, IPCA predicts the largest number of patterns, and 96% of the predicted patterns do not matched with any protein complex. At $o \geq 0.6$, the number of matched known protein complexes by *MISPAG-PCSA* is the best, MCL is the second, and IPCA is the third. *MISPAG-PCSA* matches 1.17 and 1.45 times more than MCL and IPCA. At the overlapping score, $o \geq 0.7$ to $o \geq 1.0$, the

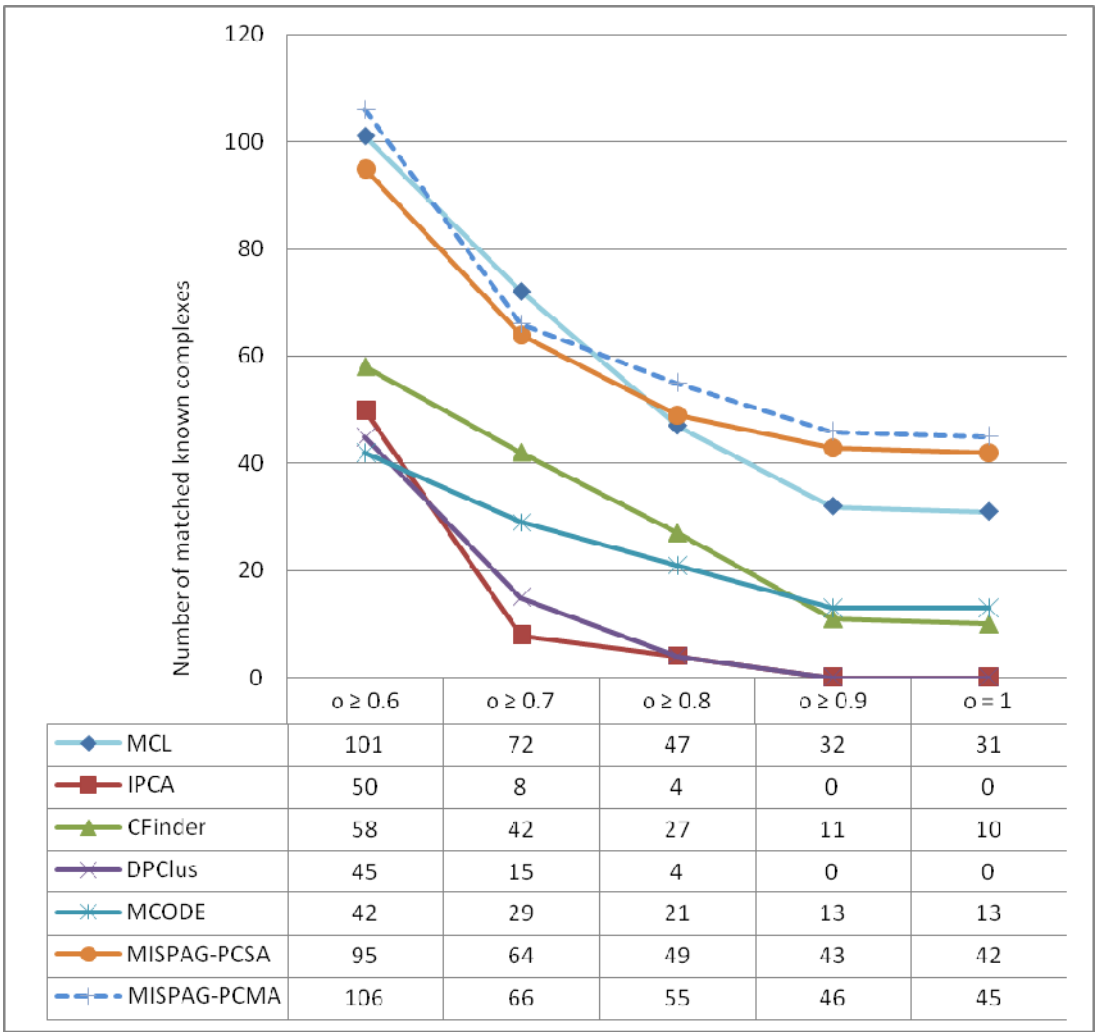
performance of *MISPAG-PCSA* is still the best among the other algorithms, MCL is the second, and CFinder is the third. The results show that *MISPAG-PCSA* outperforms all these five existing algorithms on the performance of identifying protein complexes. Comparing the performance of *MISPAG-PCSA* and MCL, not only the number of discovered protein complexes of *MISPAG-PCSA* is larger than MCL, the number of false positive patterns of *MISPAG-PCSA* is also smaller than MCL, which refers to a better result.

Table 29 Comparison of the number of predicted patterns generated by different algorithms that match known protein complexes on the dataset of Krogan

Algorithm	**Total no. of predicted patterns**	***o*** **≥ 0.6**	***o*** **≥ 0.7**	***o*** **≥ 0.8**	***o*** **≥ 0.9**	***o*** **= 1.0**
MCL	634	96 (2^{nd})	46 (2^{nd})	31 (2^{nd})	25 (2^{nd})	25 (2^{nd})
IPCA	1928	77 (3^{rd})	16	4	1	1
CFinder	113	50	35 (3^{rd})	20 (3^{rd})	14 (3^{rd})	14 (3^{rd})
DPClus	636	53	18	3	1	1
MCODE	74	31	20	14	10	10
MISPAG-PCSA	495	112 (1^{st})	64 (1^{st})	51 (1^{st})	47 (1^{st})	47 (1^{st})

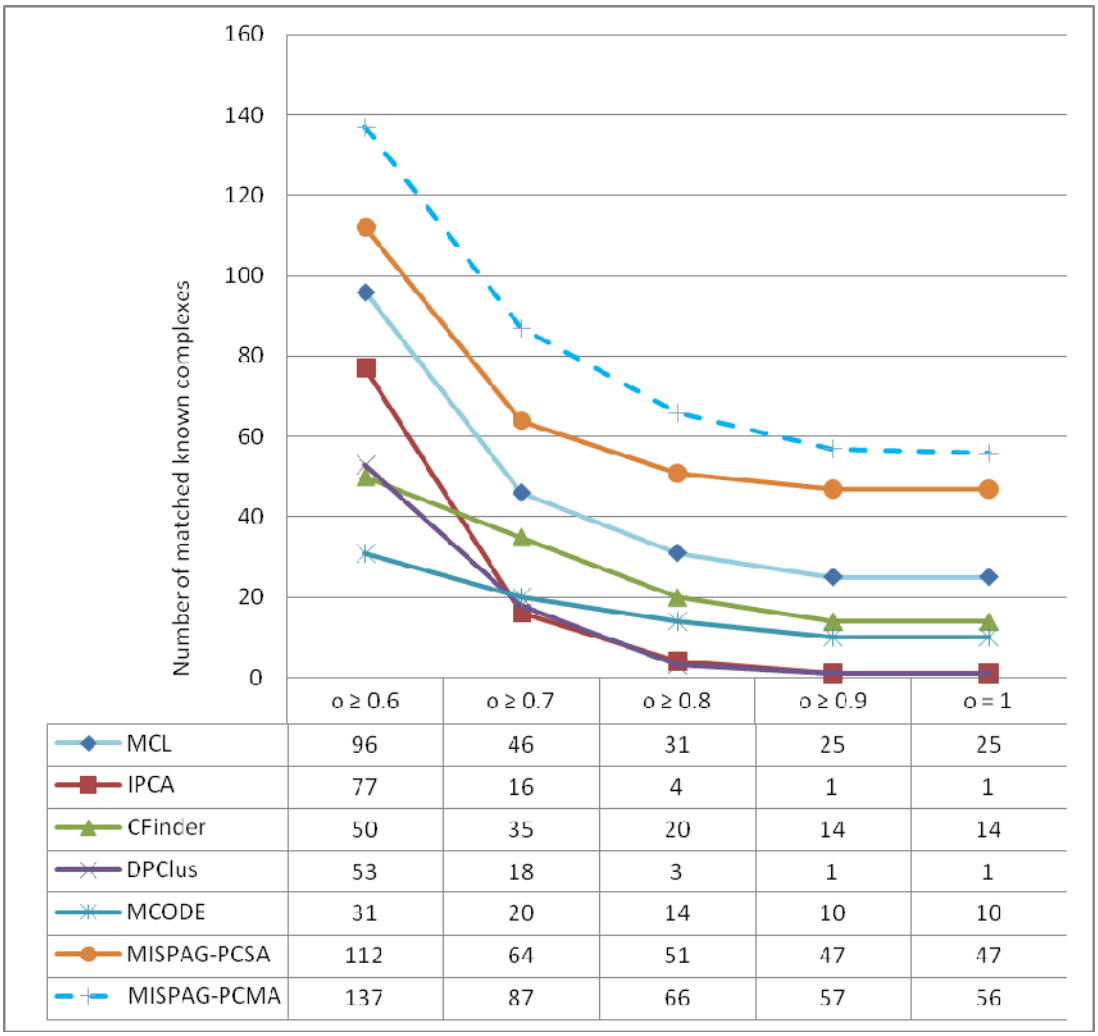
Previously, we use a single attribute with multiple values to label the PPI networks in *MISPAG-PCSA*, its results are promising. Here we try to enhance the

performance of our approach and introduce *MISPAG-PCMA* that includes multiple attributes for labeling the network. With the same parameter settings, the results of different algorithms are illustrated in *Figure* 38 and 39 for the datasets of *Gavin* and *Krogan*, respectively. The results show that *MISPAG-PCMA* work better than *MISPAG-PCSA*. For the *Gavin* dataset, *MISPAG-PCMA* is 1.03 to 1.12 times better than *MISPAG-PCSA*. At ≥ 0.6, *MISPAG-PCSA* is not as good as MCL, but now, *MISPAG-PCMA* can give a better result than MCL. For the *Krogan* dataset, *MISPAG-PCMA* is further improved with 1.19 to 1.36 times better than *MISPAG-PCSA*. It proves that our algorithm *MISPAG-PCMA* can identify much more known protein complexes than the other algorithms.



	o ≥ 0.6	o ≥ 0.7	o ≥ 0.8	o ≥ 0.9	o = 1
MCL	101	72	47	32	31
IPCA	50	8	4	0	0
CFinder	58	42	27	11	10
DPClus	45	15	4	0	0
MCODE	42	29	21	13	13
MISPAG-PCSA	95	64	49	43	42
MISPAG-PCMA	106	66	55	46	45

Figure 38 Number of matched known complexes of different algorithms with respect to different overlapping scores in the Gavin dataset



	o ≥ 0.6	o ≥ 0.7	o ≥ 0.8	o ≥ 0.9	o = 1
MCL	96	46	31	25	25
IPCA	77	16	4	1	1
CFinder	50	35	20	14	14
DPClus	53	18	3	1	1
MCODE	31	20	14	10	10
MISPAG-PCSA	112	64	51	47	47
MISPAG-PCMA	137	87	66	57	56

Figure 39 Number of matched known complexes of different algorithms with respect to different overlapping scores in the Krogan dataset

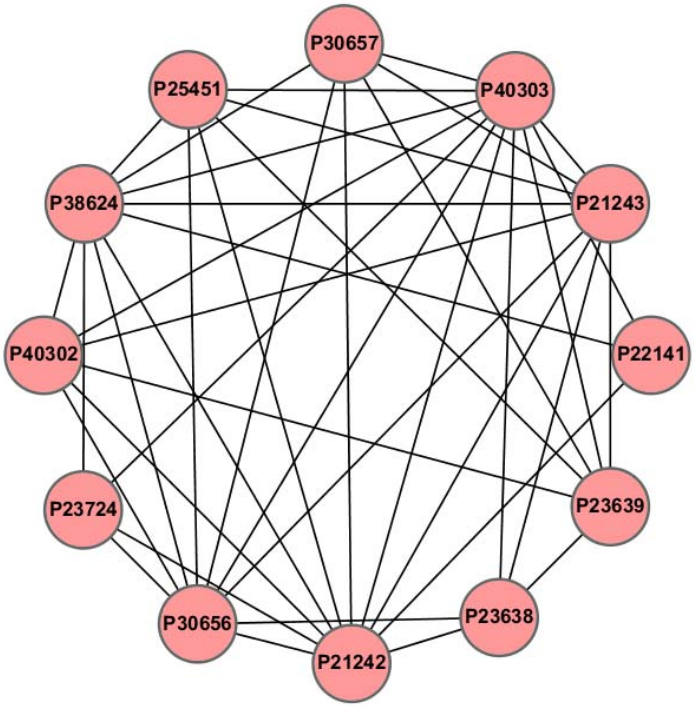
8.4.3 Analysis of Matched Protein Complexes

To understand the matched protein complexes, three predicted patterns of *MISPAG-PC* are selected for further analysis. Instead of finding densely connected regions in the PPI networks, we use *MISPAG-PC* to discover interesting interactions that are connected to form interesting patterns.

Figure 40 shows a predicted pattern of *MISPAG-PCSA* matches completely with the protein complex, *20S proteasome* (MIPS ID: 360.10.10). In this protein complex, we found that the interactions are all between the molecular function, *threonine-type endopeptidase activity*, which is found to be a very interesting interaction with *d*-score = 59.83. By calculating the interestingness score of each interaction, we can identify such interesting pattern that corresponds to a known protein complex. It confirms that the relationships between the molecular functions of proteins are useful for identifying protein complexes that are biologically meaningful patterns.

Figure 41 shows another predicted pattern of *MISPAG-PCMA* matches completely with the protein complex, TRAPP (Transport Protein Particle) complex (MIPS ID: 260.60). As the proteins in this complex all share a common biological process, *ER_to_Golgi_vesicle-mediated_transport*, and such interactions are regarded as interesting with *d*-score = 30.05, we can identify this protein complex successfully even though six out of ten proteins have unknown molecular function.

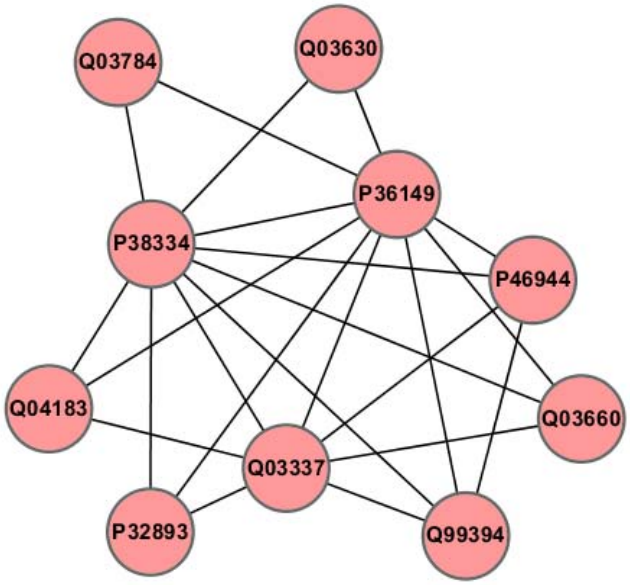
Figure 42 shows the third predicted pattern of *MISPAG-PCMA* matches 78% with the protein complex, *probably membrane biogenesis and traffic complex* (MIPS ID: 550.1.77). The predicted pattern is indicated by the dashed eclipse that contains seven proteins with different molecular functions and biological processes. We cannot detect the remaining two proteins that are supposed to be included in that protein complex may due to the problem of incompleteness in the PPI networks. Most protein complexes are inter-connected subgraphs that contain biologically related proteins, so we believe that certain interactions are existed between the two separated subgraphs.



Protein accession	Molecular functions
P21243	threonine-type endopeptidase activity
P21242	protein binding, threonine-type endopeptidase activity
P30656	protein binding, threonine-type endopeptidase activity

P38624	threonine-type endopeptidase activity
P30657	threonine-type endopeptidase activity
P40302	threonine-type endopeptidase activity
P40303	protein binding, threonine-type endopeptidase activity
P23639	protein binding, threonine-type endopeptidase activity
P23638	threonine-type endopeptidase activity
P25451	threonine-type endopeptidase activity
P22141	protein binding, threonine-type endopeptidase activity
P23724	threonine-type endopeptidase activity

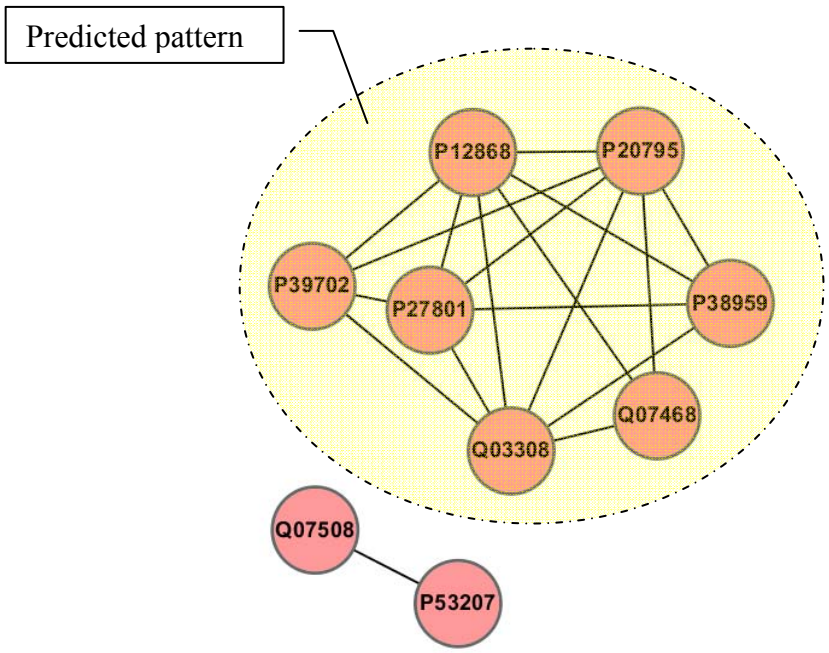
Figure 40 Matched protein complex MIPS ID: 360.10.10 with o = 1.0



Protein accession	Attribute values
P38334	protein binding, ER to Golgi vesicle
P36149	protein binding, ER to Golgi vesicle
Q03630	unknown molecular function, ER to Golgi vesicle
P46944	unknown molecular function, ER to Golgi vesicle

P32893	unknown molecular function, cell wall organization, ER to Golgi vesicle
Q04183	protein binding, ER to Golgi vesicle
Q03660	unknown molecular function, ER to Golgi vesicle
Q03784	unknown molecular function, chromosome organization, ER to Golgi vesicle
Q03337	protein binding, ER to Golgi vesicle
Q99394	unknown molecular function, ER to Golgi vesicle

Figure 41 Matched protein complex MIPS ID: 260.60 with o = 1.0



Protein accession	Attribute values
P39702	protein binding, zinc ion binding, late endosome to vacuole transport, protein transport
P27801	protein binding, zinc ion binding, Golgi to endosome transport, intracellular protein transport, late endosome to vacuole transport, response to drug, vacuole fusion, non-autophagic, vesicle docking during exocytosis
P12868	protein binding, zinc ion binding, Golgi to endosome transport, late endosome to vacuole transport, protein transport, vacuole fusion, non-autophagic, vesicle docking during exocytosis

Q03308	unknown molecular function, Golgi to endosome transport, late endosome to vacuole transport, protein targeting to vacuole, vacuole fusion, non-autophagic
P20795	ATP binding, Golgi to endosome transport, late endosome to vacuole transport, piecemeal microautophagy of nucleus, protein transport, vacuole fusion, non-autophagic, vesicle docking during exocytosis
P38959	protein binding, Rab guanyl-nucleotide exchange factor activity, intracellular protein transport, piecemeal microautophagy of nucleus, vacuolar protein processing, vacuole fusion, non-autophagic, vesicle-mediated transport
Q07468	Rab guanyl-nucleotide exchange factor activity, piecemeal microautophagy of nucleus, protein transport, vacuole fusion, non-autophagic
P53207	protein binding, RNA binding, nuclear mRNA splicing via spliceosome
Q07508	mRNA binding, mRNA splice site selection

Figure 42 Matched protein complex MIPS ID: 550.1.77 with o = 0.78

8.5 Summary

While the patterns discovered by the existing algorithms are solely based on the topology information of the PPI networks to discover protein complexes, we proposed *MISPAG-PC* that includes the biological information, molecular functions and biological processes, to identify biologically significant patterns that can match more number of known protein complexes.

The PPI networks are represented as a network graph, and labeled with single and multiple attributes based on the Gene Ontology database. By discovering interesting interactions between proteins, we can generate a set of biologically meaningful patterns from the PPI networks. To confirm these patterns are biologically meaningful, we applied our algorithm *MISPAG-PC* to two sets of PPI networks. The predicted patterns are further matched with the known protein complexes that are defined in MIPS database for performance evaluation. The MIPS

provides a benchmark collection of protein complexes, if our predicted patterns can match many of these complexes, it proves that our algorithm can discover biological patterns. The results show that both versions of our algorithm (*MISPAG-PCSA* and *MISPAG-PCMA*) can identify more protein complexes than many well-known algorithms, including MCL, IPCA, CFinder, DPClus, and MCODE. It further confirms that mining with solely structural information is not enough, and more biological information such as molecular functions and biological processes should be included for better analysis. Since the current collection of protein complexes in MIPS is still far from complete, the predicted protein complexes of *MISPAG-PC* are potentially be the real protein complexes that provide insights for biologists to explore more novel protein complexes.

Chapter 9

Conclusions

In the previous chapters, we provided a set of computational algorithms that enables researchers to uncover interesting patterns from the complex structure of biological interaction networks. The proposed methods and biological results obtained in this dissertation provide us a better understanding of the relationships between structure and function in various networks.

Large-scale biological experiments can directly detect a large amount of interactions between biomolecules, however, the resulting data sets are often incomplete and exhibit high error rates. To discover useful and interesting patterns from these networks, we need a novel graph mining approach that can uncover interesting patterns as well as robust to the noisy environment.

Many existing graph mining algorithms discover frequent subgraphs as network motifs. However, a network motif is much more than a frequently-occurred subgraph, they are not necessarily being the most frequently-occurring ones. Instead of finding frequent patterns, we introduced a novel graph mining approach called *MISPAG* that based on a residual analysis to define interestingness measure for discovering interesting subgraph patterns that possess the power of characterization and discrimination from biological interaction databases. *MISPAG* discovers interesting patterns that recur in the network and deviate from the expected probability.

We applied various versions of *MISPAG* on different biological data and applications to test its effectiveness and performance. With the discovered interesting patterns, we can classify the unknown networks to their classes from the multiple attributed networks by *MISPMAG-MA* successfully. *MISPAG-PF* is able to predict the function of un-annotated biomolecules with the discovered interesting patterns as association rules, and *MISPAG-DM* can discover some new interesting patterns in biological networks that the existing frequent graph mining algorithms cannot find. *MISPAG-FP* can work very well on filtering the uninteresting patterns that are discovered from large and complex biological data sets. *MISPAG-CM* can discover interesting subgraphs while taking into consideration both the characterization and discrimination information for molecular classification, and *MISPAG-PC* can identify biologically meaningful protein complexes from the PPI networks by considering the function relatedness between proteins.

To conclude, our proposed algorithms have the advantage that it is able to discover subgraphs that are interesting in characterizing a class and filtering those that are not. Besides, it can significantly reduce the number of subgraphs that need to be generated for network classification and function prediction by filtering those irrelevant subgraphs. As no user-defined threshold such as the use of a support and confidence measure in frequent subgraph algorithms is required, it can discover subgraphs that are relatively lower in frequency but are useful in distinguishing graphs from one class to the other. It is a better solution over the existing graph mining algorithms for network classification, prediction, and clustering tasks.

Computational learning of biological interaction networks is still a challenging research domain. Although several sub-problems such as motif discovery, protein function prediction, and protein complexes identification have been studied for a while, many important issues still remain unsolved. In this dissertation, we covered six important areas including the identification of interesting structural patterns, the discovery of interesting motifs from PPI networks, a hybrid neighbor approach for function prediction, an enhanced feature-based classification of PPI networks, the discovery of class-specific patterns from molecular data, and the identification of protein complexes with biological and structural information in PPI networks. In the future, we would like to test the adaptability of *MISPAG* by applying it to a wider variety of applications and data sets. In order to facilitate understanding, we would like represent the interesting subgraphs in a more flexible structure so that the subgraphs that are similar can be synthesized as a module, which would allow a more flexible representation of patterns, and provide a significant insight into the cause of disease as well as drug design.

References

[1] C. L. Myers, D. Robson, A. Wible, M. A. Hibbs, C. Chiriac, C. L. Theesfeld, K. Dolinski, and O. G. Troyanskaya, Discovery of biological networks from diverse functional genomic data, *Genome Biology*, 6, R114, 2005.

[2] A. W. Rives, and T. Galitski, Modular organization of cellular networks, *Proc. Natl. Acad. Sci. of U.S.A.*, 100(3), pp. 1128–33, 2003.

[3] M. Koyuturk, A. Grama, and W. Szpankowski, An efficient algorithm for detecting frequent subgraphs in biological networks, *Bioinformatics*, 20(1), pp. 200-207, 2004.

[4] B. Schwikowski, P. Uetz, and S. Fields, A network of protein–protein interactions in yeast, *Nat. Biotechnology*, 18, pp. 1257-1261, 2000.

[5] A. L. Barabasi, and Z. N. Oltvai, Network biology: understanding the cell's functional organization, *Nat. Rev. Genet.*, 5(2), pp. 101–13, 2004.

[6] R. Sharan, T. Ideker, B. Kelley, R. Shamir, and R. M. Karp, Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data, *Journal of Computational Biology*, 12(6), pp. 835-846, 2005.

[7] M. E. J. Newman, Power laws, Pareto distributions and Zipf's law, *Contemporary Physics*, 46, pp. 323-351, 2005.

[8] The Gene Ontology Consortium, The Gene Ontology (GO) database and informatics resource, *Nucleic Acids Research*, 32, 2004.

[9] M. Koyuturk, Y. Kim, S. Subramaniam, W. Szpankowski, and A. Grama, Detecting Conserved Interaction Patterns in Biological Networks, *Journal of Computational Biology*, 13(7), pp. 1299-1322, 2006.

[10] R. Chittimoori, L. B. Holder, and D. J. Cook, Applying the *Subdue* Substructure Discovery System to the Chemical Toxicity Domain, *Proceedings of the AAAI Spring Symposium on Predictive Toxicology of Chemicals: Experiences and Impact of AI Tools*, 1999.

[11] A. Inokuchi, T. Washio, T. Okada, and H. Motoda, Applying the apriori-based graph mining method to mutagenesis data analysis, *Journal of Computer Aided Chemistry*, 2, pp. 87-92, 2001

[12] K. C. C. Chan, and A. K. C. Wong, A Statistical Technique for Extracting *Class*ificatory Knowledge from Databases, *Knowledge Discovery in Databases*, G.

Piatetsky-Shapiro and W.J. Frawley, eds., Cambridge, *Mass.: AAAI/MIT Press*, pp. 107-123, 1991.

[13] P. C. H. Ma, and K. C. C. Chan, UPSEC: An Algorithm for *Class*ifying Unaligned Protein Sequences into Functional Families, *Journal of Computational Biology*, 15(4), pp.431-443, 2008.

[14] P. Pagel, S. Kovac, M. Oesterheld, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, P. Mark, V. Stümpflen, H. W. Mewes, A. Ruepp, and D. Frishman, The MIPS mammalian protein-protein interaction database, *Bioinformatics*, 21(6), pp. 832-834, 2005.

[15] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, The Database of interacting Proteins: 2004 update, *Nucleic Acids Research*, 32, D449-51, 2004.

[16] B. J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. Lackner, J. Bahler, V. Wood, K. Dolinski, and M. Tyers, The BioGRID Interaction Database: 2008 Update, *Nucleic Acids Research*, 2008.

[17] L. J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic, P. Bork, C. von Mering, STRING 8--a global view on proteins and their functional interactions in 630 organisms, *Nucleic Acids Research*, 2009.

[18] The UniProt Consortium, The Universal Protein Resource (UniProt), Nucleic Acids Res. 36:D190-D195, 2008.

[19] S. Daskalaki, I. Kopanas, N. Avouris, Evaluation of *Class*ifiers for an Uneven class Distribution Problem, *Applied Artificial Intelligence*, 20(5), pp. 381-417, 2006.

[20] T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters*, 27, pp. 861-874, 2006.

[21] H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi, Assessment of prediction accuracy of protein function from protein--protein interaction data, *Yeast*, 18(6), pp. 523-31, 2001.

[22] C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Guenoche, and B. Jacq, Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network, *Genome Biology*, 5: R6, 2003.

[23] L. B. Holder, D. J. Cook, and S. Djoko, Substructure Discovery in the *SUBDUE* System, *In Proceedings of the AAAI Workshop on Knowledge Discovery in Databases*, pp. 169-180, 1994.

[24] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, Network motifs: Simple building blocks of complex networks, *Science*, 298(5594), pp. 824-827, 2002.

[25] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, From molecular to modular cell biology, *Nature*, 402, c47– c52, 1999.

[26] L. R. Matthews, P. Vaglio, J. Reboul, H. Ge, B. P. Davis, J. Garrels, S. Vincent, and M. Vidal, Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or 'interologs', *Genome Research*, 11, pp. 2120–2126, 2001.

[27] B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker, Conserved pathways within bacteria and yeast as revealed by global protein network alignment, *Proc. Natl Acad. Sci. USA*, 100, pp. 11394–11399, 2003.

[28] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, Basic local alignment search tool, *Journal of Molecular Biology*, 215, pp. 403-410, 1990.

[29] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker, Conserved patterns of protein interaction in multiple species, *Proceedings of the National Academy of Sciences of USA*, 102 (6), pp. 1974–1979, 2005.

[30] M. Koyuturk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, A. Grama, Pairwise alignment of protein interaction networks, *J Comput Biol.*, 13, pp. 182–199, 2006.

[31] J. Reimand, L. Tooming, H. Peterson, P. Adler, and J. Vilo, GraphWeb: mining heterogeneous biological networks for gene modules with functional significance, *Nucleic Acids Research*, 36, W452-9, 2008.

[32] A. L. Barabasi, and Z. N. Oltvai, Network biology: understanding the cell's functional organization*, Nat. Rev. Genet.*, 5(2), pp. 101–13, 2004.

[33] Rives, A. W., and Galitski, T., Modular organization of cellular networks, *Proc. Natl. Acad. Sci. U.S.A.,* 100 (3), pp.1128–33, 2003.

[34] M. Koyuturk, A. Grama, and W. Szpankowski, An efficient algorithm for detecting frequent subgraphs in biological networks, *Bioinformatics*, 20(1), pp. 200-207, 2004.

[35] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, Network motifs in the transcriptional regulation network of Escherichia coli, *Nature Genetics*, 31, pp. 64-68, 2002.

[36] P. Aloy, and R. B. Russell, Interrogating protein interaction networks through structural biology. *Proc Natl Acad Sci USA*, 99, pp. 5896–901, 2002.

[37] F. Schreiber, and H. Schwobbermeyer, Frequency concepts and pattern detection for the analysis of motifs in networks, *Trans Comput Syst Biol III*, LNBI, 3737, pp. 89–104, 2005.

[38] L. Parida, Discovering Topological motifs using a compact notation. *J Comput Biol*, 4(3), pp. 300–23, 2007.

[39] A. Inokuchi, T. Washio, and H. Motoda, An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data, *Principles of Data Mining and Knowledge Discovery*, pp. 13-23, 2000.

[40] M. Kuramochi, and G. Karypis, Frequent Subgraph Discovery, *First IEEE International Conference on Data Mining (ICDM'01)*, pp. 313, 2001.

[41] M. E. Turanalp, and T. Can, Discovering functional interaction patterns in protein-protein interaction networks, *BMC Bioinformatics*, 9:276, 2008.

[42] C. Chen, X. Yan, F. Zhu, and J. Han, gApprox: Mining Frequent Approximate Patterns from a Massive Network, *The Seventh IEEE International Conference on Data Mining (ICDM '07),* pp.445-450, 2007.

[43] Brohee S. and Helden J. (2006) Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics, **7**, 488.

[44] M. Kuramochi, and G. Karypis, An Efficient Algorithm for Discovering Frequent Subgraphs, *IEEE Trans. Knowl. Data Eng*., 16(9), pp. 1038-1051, 2004.

[45] X. Yan, and J. Han, *gSpan*: Graph-based substructure pattern mining, *Proceedings of IEEE International Conference on Data Mining ICDM*, pp. 721-724, 2002.

[46] H. Chernoff, and E. L. Lehmann, The use of maximum likelihood estimates in $\chi 2$ tests for goodness-of-fit, *The Annals of Mathematical Statistics*, 25(3), pp. 579-586, 1954.

[47] M. P. Samanta, and S. Liang, Predicting protein functions from redundancies in large-scale protein interaction networks, *Proc Natl Acad Sci U S A.,* 100(22), pp. 12579-12583, 2003.

[48] A. Berkopec, HyperQuick algorithm for discrete hypergeometric distribution, Journal of Discrete Algorithms, *Elsevier*, 2006.

[49] C. Brun, C. Herrmann, and A. Guenoche, Clustering proteins from interaction networks for the prediction of cellular functions, *BMC Bioinformatics*, 5, 95, 2004.

[50] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis, Frequent Substructure-Based Approaches for *Class*ifying Chemical Compounds, *IEEE Transactions on Knowledge and Data Engineering*, 17 (8), pp. 1036-1050, 2005.

[51] N. Wale, and G. Karypis, Acyclic Subgraph-based Descriptor Spaces for Chemical Compound Retrieval and *Class*ification, *In Proc of IEEE International Conference on Data Mining (ICDM),* 2006.

[52] T. Horvath, T. Grtner, and S. Wrobel, Cyclic pattern kernels for predictive graph mining, *In Proc. of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 158-167, 2004.

[53] T. Kudo, E. Maeda, and Y. Matsumoto, An Application of Boosting to Graph *Class*ification, *NIPS*, 2004.

[54] T. Joachims, Making large-Scale *SVM* Learning Practical, *Advances in Kernel Methods - Support Vector Learning*, B. Scholkopf and C. Burges and A. Smola (ed.), *MIT Press*, 1999.

[55] C.Z Cai., W.L. Wang, L.Z. Sun, and Y.Z. Chen, Protein function classification via support vector machine approach, *Math. Biosci.*, 185, pp. 111–122, 2003.

[56] P.D. Dobson, and A.J. Doig, Distinguishing enzyme structures from non-enzymes without alignments. J. Mol. Biol., 330, pp. 771–783, 2003.

[57] S. Tornow, and H. W. Mewes, Functional modules by relating protein interaction networks and gene expression, *Nucleic Acids Research*, 31(21), pp. 6283–6289, 2003.

[58] S. Milgram, The small-world problem, *Psychology Today*, 1, pp. 61-67, 1967.

[59] D. J. Watts, and S. H. Strogatz, Collective dynamics of ‘small-world’ networks, *Nature*, 393, pp. 440-442, 1998.

[60] A.L. Barabasi, Emergence of scaling in complex networks, *Handbook of Graphs and Networks*, Ed. S. Bornholdt and H.G. Schuster, pp. 69-84, *Wiley-VCH*, Weinheim, Germany, 2003.

[61] W. McGinnis, R. L. Garber, J. Wirz, A. Kuroiwa, and W. J. Gehring, A homologous protein-coding sequence in Drosophila homeotic genes and its conservation in other etazoans, *Cell*, 37:403, 1984.

[62] I. D. Campbell, A. K. Downing, Building protein structure and function from modular units, *Trends Biotechnol*, 12 (5), pp.168-72, 1994.

[63] C. Lin, D. Jiang,and A. Zhang, Prediction of Protein Function Using Common-Neighbors in Protein-Protein Interaction Networks, *Proceedings of the Sixth IEEE Symposium on BionInformatics and BioEngineering (BIBE),* pp.251-260, 2006.

[64] G. Bader, and C. Hogue, Analyzing yeast protein-protein interaction data obtained from different sources, *Nature Biotechnology,* 20, pp. 991-997, 2002.

[65] C. Von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork, Comparative assessment of large-scale data sets of protein-protein interactions, *Nature*, 417(6887), pp. 399-403, 2002.

[66] H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai, Lethality and centrality in protein networks, *Nature*, 411, pp. 41-42, 2001.

[67] Z. Lu, and L. Hunter, GO Molecular Function Terms Are Predictive of Subcellular Localization, *Pacific Symposium on Biocomputing*, 10, pp.151-161, 2005.

[68] Andrew Chatr-aryamontri; Arnaud Ceol; Luisa Montecchi Palazzi; Giuliano Nardelli; Maria Victoria Schneider; Luisa Castagnoli; Gianni Cesareni, MINT: the Molecular INTeraction database, *Nucleic Acids Research*, 2006.

[69] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thorneycroft, Y. Zhang, R. Apweiler, H. Hermjakob, IntAct – Open Source Resource for Molecular Interaction Data, *Nucleic Acids Research*, 2006.

[70] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi, KEGG for linking genomes to life and the environment, *Nucleic Acids Research*, 36, D480-D484, 2008.

[71] X. M. Zhao, Y. Wang, L. Chen, and K. Aihara, Gene function prediction using labeled and unlabeled data, *BMC Bioinformatics*, 9:57, 2008.

[72] K. Jahn, and S. Kramer, Optimizing *gSpan* for Molecular Datasets, *Proceedings of the Third International Workshop on Mining Graphs, Trees and Sequences* (MGTS-2005), 2005.

[73] N. Lin, B. Wu, R. Jansen, M. Gerstein, and H. Zhao, Information assessment on predicting protein-protein interactions, *BMC Bioinformatics*, 5, pp.154, 2004.

[74] S.H. Muggleton, Inductive Logic Programming, *New Generation Computing*, 8(4), pp.295-318, 1991.

[75] R. D. King, A. Srinivasan, and L. Dehaspe, Warmr: a data mining tool for chemical data, *Journal of Computer-Aided Molecular Design*, 15(2), pp.173-181, 2001.

[76] T. S. K. Prasad, Human Protein Reference Database - 2009 Update, *Nucleic Acids Research*, doi:10.1093/nar/gkn892, 2008.

[77] Caspi et al., The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases, *Nucleic Acids Research*, 2008.

[78] Matthews L, Gopinath G, Gillespie M, Caudy M, Croft D, de Bono B, Garapati P, Hemish J, Hermjakob H, Jassal B, Kanapin A, Lewis S, Mahajan S, May B, Schmidt E, Vastrik I, Wu G, Birney E, Stein L, D'Eustachio P. Reactome knowledgebase of biological pathways and processes, *Nucleic Acids Res.,* 2008.

[79] A. Fader, G. Erkan, A. Ozgur, A. Ade, J. Gerrish, M. Schaller, D. States, and D. Radev. GIN: Gene Interaction Network, *NCIBI All-Hands Meeting* Poster Session, 2007.

[80] E.A. Ananko, N.L. Podkolodny, I.L. Stepanenko, E.V. Ignatieva, O.A. Podkolodnaya, N.A. Kolchanov, GeneNet: a database on structure and functional organisation of gene networks, *Nucleic Acids Res.*, 30 (1), pp.398-401, 2002.

[81] G. Zheng, K. Tu, Q. Yang, Y. Xiong, C. Wei, L. Xie, Y. Zhu, and Y. Li, ITFP: an integrated platform of mammalian transcription factors, *Bioinformatics*, 24(20):2416-2417, 2008.

[82] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N.J. Krogan, S. Chung, A. Emili, M. Snyder, J.F. Greenblatt, M. Gerstein, A Bayesian networks approach for predicting protein-protein interactions from genomic data, *Science*, 302, pp. 449-453, 2003.

[83] A. L. Barabsi, and R. Albert, Emergence of Scaling in Random Networks, *Science,* New Series, 286 (5439), pp. 509-512, 1999.

[84] K. I. Goh, E. Oh, H. Jeong, B. Kahng, and D. Kim, *Class*ification of scale-free networks, *Proceedings of the National Academy of Sciences*, 99(20), pp.12583-12588, 2002.

[85] L. Hamill, and N. Gilbert, Social Circles: A Simple Structure for Agent-Based Social Network Models, *Journal of Artificial Societies and Social Simulation*, 12(2)3, 2009.

[86] PubMed Central (PMC), U.S. National Institutes of Health (NIH), http://www.pubmedcentral.nih.gov/, 2009.

[87] A. Birkland, and G. Yona, Biozon: a system for unification, management and analysis of heterogeneous biological data, *BMC Bioinformatics*, 70(7), 2006.

[88] R. Stevens, C. Goble, N. Paton, S. Bechhofer, G. Ng, P. Baker, and A. Brass, Complex query formulation over diverse information sources in tambis, *Bioinformatics: Managing Scientific Data*, 2003.

[89] J. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, Advances in Kernel Methods - Support Vector Learning, *MIT Press*, Cambridge, MA, 1998.

[90] I. Witten, and E. Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, *Morgan Kaufmann*, San Francisco, CA, 2005.

[91] A. J. Enright, V. Kunin, and C. A. Ouzounis, Protein families and TRIBES in genome sequence space, *Nucleic Acids Research*, 31(15), pp. 4632-4638, 2003.

[92] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, A comprehensive two hybrid analysis to explore the yeast protein interactome, In Proceedings of *National Academy of Sciences* USA, 98, pp. 569–4574, 2001.

[93] A. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J.M. Rick, A. Michon, and C. Cruciat, Functional organization of the yeast proteome by systematic analysis of protein complexes, *Nature*, 415, pp. 141–147, 2002.

[94] B. Schweitzer, P. Predki, and M. Snyder, Microarrays to characterize protein interactions on a whole proteome scale, *Proteomics*, 3, pp. 2190–2199, 2003.

[95] A.H. Tong, M. Evangelista, A.B. Parsons, H. Xu, and G.D. Bader, Systematic genetic analysis with ordered arrays of yeast deletion mutants, *Science*, 294, pp. 2364–2368, 2001.

[96] H. Ogata, W. Fujibuchi, S. Goto, and M. Kanehisa, A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters, *Nucleic Acids Res.*, 28, pp. 4021–4028, 2000.

[97] S. Mangan, A. Zaslaver, and U. Alon, The Coherent Feedforward Loop Serves as a Signsensitive Delay Element in Transcription Networks, *J. Mol. Biol.*, 334, pp. 197-204, 2003.

[98] Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U., Network motifs in the transcriptional regulation network of Escherichia coli, *Nature Genetics*, 31, pp. 64-68, 2002.

[99] A. L. Barabasi, and Z. N. Oltvai, Network Biology: Understanding the Cell's Functional Organization, Nature Reviews, Genetics, 5, pp.101-113, 2004.

[100] O. Keskin, R. L. Jernigan, and I. Bahar, Proteins with Similar Architecture Exhibit Similar Large-Scale Dynamic Behavior, Biophysical Journal, 78, pp. 2093-2106, 2000.

[101] K. Takabayashi, P. C.Nguyen, K. Ohara, H. Motoda, and T. Washio, Mining Discriminative Patterns from Graph Structured Data with Constrained Search, *Proceedings of Mining and Learning with Graphs* (MLG), pp.205 -212, 2006.

[102] W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. Yu, and O. Verscheure, Direct mining of discriminative and essential frequent patterns via model-based search tree, *KDD '08*, Las Vegas, Nevada, USA, 2008.

[103] O. Meshi, T. Shlomi, and E. Ruppin, Evolutionary conservation and over-representation of functionally enriched network patterns in the yeast regulatory network, *BMC Systems Biology*, 1, 1, 2007.

[104] J. L. Badano, and N. Katsanis, Beyond Mendel: an evolving view of human genetic disease transmission, *Nat. Rev. Genet*., 3, pp. 779–789, 2002.

[105] H. G. Brunner, and M. A. van Driel, From syndrome families to functional genomics, *Nat. Rev. Genet.*, 5, pp. 545–551, 2004.

[106] P. F. Jonsson, P. A. Bates, Global topological features of cancer proteins in the human interactome, *Bioinformatics*, 22, pp.2291–7, 2006.

[107] P. Uetz, Y. A. Dong, C. Zeretzke, Herpesviral protein networks and their interaction with the human proteome, *Science*, 311, pp.239–42, 2006.

[108] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, Network motifs in the transcriptional regulation network of Escherichia coli, *Nature Genetics*, 31, pp. 64-68, 2002.

[109] G. Liu, L. Wong, and H. N. Chua, Complex discovery from weighted PPI networks, *Bioinformatics*, 25(15), pp. 1891-1897, 2009.

[110] A. K. Bjorklund, S. Light, L. Hedin, and A. Elofsson, Quantitative assessment of the structural bias in protein-protein interaction assays, *Proteomics*, 8(22), pp. 4657-4667, 2008.

[111] G. T. Hart, I. Lee, and E. M. Marcotte, A high-accuracy consensus map of yeast protein complexes reveals modular nature of gene essentiality, *BMC Bioinformatics*, 8, pp. 236, 2007

[112] A. C. Gavin et al., Proteome survey reveals modularity of the yeast cell machinery, *Nature*, 440, 631–636, 2006.

[113] N. J. Krogan et al., Global landscape of protein complexes in the yeast Saccharomyces cerevisiae, *Nature*, 440, pp. 637-643, 2006.

[114] S. Nijssen, and J. N. Kok, "Frequent Graph Mining and its Application to Molecular Databases," In W. Thissen, P. Wieringa, M. Pantic, M. Ludema eds.: *Proc. of the 2004 IEEE Conf. on Systems, Man and Cybernetics*, SMC 2004, Den Haag, The Netherlands, pp. 4571 – 4577, 2004.

[115] M. Worlein, T. Meinl, I. Fischer, and M. Philippsen, "A quantitative comparison of the subgraph miners MoFa, *gSpan*, FFSM, and Gaston," In: A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama (Eds.): Knowledge Discovery in Database: PKDD 2005 (*9th European Conference on Principles and Practices of Knowledge Discovery in Databases*, Porto, Portugal). Berlin: Springer, S. 392-403, *Lecture Notes in Computer Science*, 2005.

[116] C. Borgelt, and M. R. Berthold, "Mining Molecular Fragments: Finding Relevant Substructures of Molecules," *Second IEEE International Conference on Data Mining* (ICDM'02), pp.51-58, 2002.

[117] I. Fischer, and T. Meinl, "Graph based molecular data mining - an overview," Systems, Man and Cybernetics, *IEEE International Conference*, 5, pp. 4578-4582, 2004.

[118] FSG [Online]. Available: http://www-users.cs.umn.edu/~karypis/pafi/

[119] gSpan [Online]. Available: http://illimine.cs.uiuc.edu/download/index.php

[120] C. C. Chang, and C. J. Lin. (2001) "LIBSVM: a library for support vector machines" [Online] Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[121] S. Daskalaki, I. Kopanas, and N. Avouris, "Evaluation of Classifiers for an Uneven class Distribution Problem," *Applied Artificial Intelligence*, 20(5), pp. 381-417, 2006.

[122] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, 27, pp. 861-874, 2006.

[123] A. Srinivasan, R. D. King, S. H. Muggleton, and M. Sternberg, “The predictive toxicology evaluation challenge”, *15th IJCAI*, 1997.

[124] S. Menchetti, F. Costa, and P. Frasconi, “Weighted decomposition kernels,” *Proceedings of the 22nd international conference on Machine learning*, Bonn, Germany, pp. 585-592, 2005.

[125] W. Tong, H. Fang, C. R. Williams, J. M. Burch, and A. M. Richard. (2008) “DSSTox FDA National Center for Toxicological Research Estrogen Receptor Binding Database (NCTRER): SDF files and website documentation,” NCTRER_v4b_232_15Feb2008 [Online]
Available:, www.epa.gov/ncct/dsstox/sdf_nctrer.html.

[126] M. Deshpande, and G. Karypis, “Automated approaches for classifying structure,” *Proc. of the 2nd ACM SIGKDD Workshop on Data Mining in Bioinformatics*, pp.11-18, 2002.

[127] M. Worlein, T. Meinl, I. Fischer, and M. Philippsen, “A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston,” In: A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama (Eds.): Knowledge Discovery in Database: PKDD 2005 (*9th European Conference on Principles and Practices of Knowledge Discovery in Databases*, Porto, Portugal). Berlin: Springer, S. 392-403, *Lecture Notes in Computer Science*, 2005.

[128] R Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, Chklovskii D., Alon U: Network motifs: Simple building blocks of complex networks, *Science* 2002, 5594(298):824–827.

[129] Adamcsek B, Palla G, Farkas I, Der’enyi I, Vicsek T: CFinder: locating cliques and overlapping modules in biological networks, *Bioinformatics* 2006, 22:1021–1023.

[130] Wuchty S, Oltvai ZN, Barabasi AL: Evolutionary conservation of motif constituents in the yeast protein interaction network, *Nature Genetics* 2003, 35(2):176-179.
[131] Pereira-Leal JB, Enright AJ, Ouzounis CA, Detection of functional modules from protein interaction networks, *Proteins* 2004, 54:49–57.

[132] Vazquez A, Flammini A, Maritan A, Vespignani A: Global protein function prediction from protein-protein interaction networks, *Nat Biotechnol* 2003, 21(6):697–700.

[133] Brun C, Chevenet F, Martin D, Wojcik J, Guenoche A, Jacq B: Functional classification of proteins for the prediction of cellular function from a protein–protein interaction network, *Genome Biol.* 2003, 5:R6.

[134] Spirin V, Mirny LA: Protein complexes and functional modules in molecular networks, *Proc Natl Acad Sci U S A* 2003, 100:12123–12128.

[135] Hirsh E, Sharan R: Identification of conserved protein complexes based on a model of protein network evolution, *Bioinformatics* 2007, 23(2):e170-e176.

[136] Chen J, Hsu W, Lee ML, Ng SK: Labeling network motifs in protein interactomes for protein function prediction*, IEEE 23rd International Conference on Data Engineering*, ICDE 2007, 546-555.

[137] Auerbach D, Arnoldo A, Bogdan B, Fetchko M, Stagljar I: Drug Discovery Using Yeast as a Model System: A Functional Genomic and Proteomic View, *Current Proteomics* 2005, 2:1-13.

[138] Chu LH, Chen BS: Construction of a cancer-perturbed protein-protein interaction network for discovery of apoptosis drug targets, *BMC System Biology* 2008, 2:56.

[139] Kelley BP, Sharan R, Karp RM, Sittler T, Root DE, Stockwell BR, Ideker T: Conserved pathways within bacteria and yeast as revealed by global protein network alignment, *Proc. Natl. Acad. Sci. U S A* 2003, 100(20):11394–11399.

[140] Sharan R, Suthram S, Kelley RM, Kuhn T, McCuine S, Uetz P, Sittler T, Karp RM, Ideker T: Conserved patterns of protein interaction in multiple species, *Proceedings of the National Academy of Sciences of USA* 2005, 102(6):1974–1979.

[141] Koyuturk M, Kim Y, Topkara U, Subramaniam S, Szpankowski W, Grama A: Pairwise alignment of protein interaction networks, *J. Comput. Biol.* 2006, 13(2):182–199.

[142] Koyuturk M, Kim Y, Subramaniam S, Szpankowski W, Grama A: Detecting Conserved Interaction Patterns in Biological Networks, *Journal of Computational Biology* 2006, 13(7): 1299-1322.

[143] Chen C, Yan X, Zhu F, Han J: gApprox: Mining Frequent Approximate Patterns from a Massive Network, *The Seventh IEEE International Conference on Data Mining* 2007, 445-450.

[144] Yeger-Lotem E, Sattath S, Kashtan N, Itzkovitz S, Milo R, Pinter RY, Alon U, Margalit H: Network motifs in integrated cellular networks of transcription-

regulation and protein-protein interaction, *Proceedings of the National Academy of Sciences of USA* 2004, 101(16):5934-5939.

[145] Turanalp M, Can T: Discovering functional interaction patterns in protein-protein interaction networks, *BMC Bioinformatics* 2008, 9:276

[146] Borgwardt KM, Kriegel HP, Vishwanathan SVN, Schraudolph N: Graph kernels for disease outcome prediction from protein-protein interaction networks. In Altman RB, Dunker AK, Hunter L, Murray T, Klein T E, editors, *Proceedings of the Pacific Symposium of Biocomputing* 2007, Maui Hawaii, World Scientific.

[147] Ashburner M., et al.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet*. 2000, 25(1), 25-9.

[148] Martin S, Roe D, Faulon JL: Predicting protein-protein interactions using signature products, *Bioinformatics* 2005, 21(2):218-226.

[149] Middendorf M, Ziv E, Adams C, Hom J, Koytcheff R, Levovitz C, Woods G, Chen L, Wiggins C: Discriminative topological features reveal biological network mechanisms, *BMC Bioinformatics* 2004, 5:181.

[150] A. L. Barabasi, and Z. N. Oltvai: Network biology: understanding the cell's functional organization, *Nat. Rev. Genet.,*2004, 5(2): 101–13

[151] A. W. Rives, and T. Galitski: Modular organization of cellular networks, *Proc. Natl. Acad. Sci. of U.S.A.*, 2003, 100(3): 1128–33

[152] M. Koyuturk, A. Grama, and W. Szpankowski: An efficient algorithm for detecting frequent subgraphs in biological networks, *Bioinformatics*, 2004, 20(1): 200-207

[153] Yoshida,Y., Ohta,Y., Kobayashi,K., Yugami,N. (2003) Mining Interesting Patterns Using Estimated Frequencies from Subpatterns and Superpatterns, Lecture *Notes in Computer Science*, 2843, 494-501.

[154] Jones S. and Thronton J.M. (1996) Principles of protein-protein interactions. *Proc. Natl. Acad. Sci*. USA, 93, 13-20.

[155] Kumar A. and Snyder, M. (2002) Protein complexes take the bait. *Nature*, 415, 123-124.
[156] Ruepp A. et al. (2009) CORUM: the comprehensive resource of mammalian protein complexes – 2009. *Nucleic Acids Research*, doi:10.1093/nar/gkp914.

[157] Kim P.M. et al. (2006) Relating three-dimensional structures to protein networks provides evolutionary insights. *Science*, 314, 1938–1941.

[158] Bader G.D. and Hogue C.W. (2003) An automated method for finding molecular complex in large protein interaction networks. *BMC Bioinformatics*, 4, 2.

[159] Dongen, S. (2000) A cluster algorithm for graphs. *National Research Institute for Mathematics and Computer Science* in the Netherlands, Amsterdam, Technical report INS-R0010.

[160] Enright A.J. (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30(7), 1575-1584.

[161] Altaf-Ul-Amin M., et al. (2006) Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinformatics*, 7, 207.

[162] Li M., Chen J., Wang J.X., Hu B. and Chen G. (2008) Modifying the DPclus algorithm for identifying protein complexes based on new topological structures. *BMC Bioinformatics*, 9:398.

[163] Tong A.H. et al. (2002) A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, 295:321-324.

[164] Derenyi I. et al. (2005) Clique percolation in random networks. *Phys. Rev. Lett.*, 94, 160-202.

[165] Pearson K. (1905) The problem of the Random Walk, *Nature*, 72, 294.