GRAPH PATTERN MINING TECHNIQUES TO IDENTIFY POTENTIAL MODEL ORGANISMS

A Dissertation Presented

by

Ahmed Ragab Nabhan

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fullfillment of the Requirements
for the Degree of Doctor of Philosophy
Specializing in Computer Science

May, 2014

Accepted by the Faculty of the Graduate College, The University of Vermont, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, specializing in Computer Science.

Dissertation Examination Committee:

Indra Neil Sarkar, Ph.D. — Advisor

Jeffrey Bond, Ph.D.

Margaret Eppstein, Ph.D.

Peter Sheridan Dodds, Ph.D. — Chairperson

Cynthia J. Forehand, Ph.D. — Dean, Graduate College

Date: December 13, 2013

Abstract

Recent advances in high throughput technologies have led to an increasing amount of rich and diverse biological data and related literature. Model organisms are classically selected as subjects for studying human disease based on their genotypic and phenotypic features. A significant problem with model organism identification is the determination of characteristic features related to biological processes that can provide insights into the mechanisms underlying diseases. These insights could have a positive impact on the diagnosis and management of diseases and the development of therapeutic drugs. The increased availability of biological data presents an opportunity to develop data mining methods that can address these challenges and help scientists formulate and test data-driven hypotheses.

In this dissertation, data mining methods were developed to provide a quantitative approach for the identification of potential model organisms based on underlying features that may be correlated with disease manifestation in humans. The work encompassed three major types of contributions that aimed to address challenges related to inferring information from biological data available from a range of sources. First, new statistical models and algorithms for graph pattern mining were developed and tested on diverse genres of data (biological networks, drug chemical compounds, and text documents). Second, data mining techniques were developed and shown to identify characteristic disease patterns (disease fingerprints), predict potentially new genetic pathways, and facilitate the assessment of organisms as potential disease models. Third, a methodology was developed that combined the application of graph-based models with information derived from natural language processing methods to identify statistically significant patterns in biomedical text. Together, the approaches developed for this dissertation show promise for summarizing the information about biological processes and phenomena associated with organisms broadly and for the potential assessment of their suitability to study human diseases.

in memory of

my beloved mother, Fatimah (1951-2012)

Acknowledgements

This dissertation is an outcome of nearly four years of Ph.D. study made possible thanks to many wonderful people for whom I extend my sincere gratitude.

First and foremost, this dissertation research could not have been possible without the support of my family. To my mother, Fatimah, whom I lost along the way - thank you for all the sacrifices you made to bring me to this level. To my father Ragab, from whom I learned persistence and hard work. To my sisters Eman, Doaa, and Rehab, and my brother, Kareem - I really appreciate your encouragement. To my wonderful wife Hebatullah - I owe you all gratitude for your encouragement and support, which made it possible for me to complete this work. You have always been there for me. To my little lovely sons, Hamza and Ali - I love you. To all of you - thank you for always being there, cheering me up and standing behind me.

I also express my hearty gratitude to all of my friends in the Department of Computer Science and the Center for Clinical and Translational Sciences at the University of Vermont for everything they did to help me. My appreciation to all of my friends of Vermont communities: the Islamic Society of Vermont, community of the University of Vermont Apartment and Family Housing, and other communities from whom I learned voluntary work, giving to our communities, and to support one another. I share with my friends many pleasant memories about hiking, camping and other exciting activities in lovely Vermont.

Many thanks to the members of my studies and thesis committee, Dr. Jeffrey Bond, Dr. Margaret Eppstein, and Dr. Peter Dodds - thank you for your insight and advice on my research.

Finally, I sincerely thank my advisor, Dr. Neil Sarkar, for all the time, effort, and support he has given me from the beginning to the end - I am grateful to you for always

being available and willing to help in research with passion and enthusiasm. You always welcome students' ideas and your encouragement, support, and dedication brought me to the level of Ph.D. candidacy. I also thank Dr. Elizabeth Chen, Dr. Sarkars wife, for her kind help for me and other students at the Center for Clinical and Translational Sciences. Her presence in research meetings was of great help to me and other students to promote our communication and presentation skills.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation, Scientific Question, and Goals

The availability of large biological data and knowledge bases provides tremendous opportunities for gaining data-driven insights into complex biological systems, with a potential impact on human health. A high level of noise and heterogeneity characterize biological data (Myers and Troyanskaya 2007). Large volumes of biological data available in varying forms (sequences, graphs, and texts) can pose certain challenges when seeking new findings. First, large datasets present data management and curation problems such as data organization, access, and interpretation. Second, the identification of relevant statistical regularities and dependencies in high dimensional and diverse biological data resources presents a major challenge. Finding frequent patterns, gaining insight into the structure of data, and data summarization present problems related to large and high dimensional data. These problems are beyond capabilities of direct data management methods (e.g., simple

search). To tackle this challenge, there is a need for data mining methods that can integrate knowledge into predictive models that can utilize high dimensional data to test hypotheses about biological systems.

This thesis is focused on the application of graph data mining methods for gathering evidence from graph and text datasets pertaining to a specific challenge: the assessment of organisms as potential disease models. Biological processes (or mechanisms) that underlie diseases have biological entities (genes/proteins/molecules) that interact to accomplish a certain function. The interaction of these entities leads to the emergence of interesting patterns that differ according to the wiring (or linking) of entities as well as the function (or biological role) of each entity. This thesis is built on the hypothesis that functional and structural patterns of interactions between biological entities correlate with disease classes. The use of graph mining techniques reveals a certain set of patterns, termed disease fingerprints, which can be identified for a given disease class. It was also hypothesized that the fingerprints of a given disease class might be more similar to patterns in a biological interaction network for a given organism and less similar to other those of other organisms. The primary goal of the thesis was thus to develop methods to match disease fingerprints to interaction networks of organisms, with the aim to rank relevance of model organisms for a particular disease class.

Biomedical literature provides documentation of knowledge in the medical and life sciences (De Bruijn and Martin 2002). The development of biomedical literature databases (e.g., MEDLINE) and advances in natural language processing methods provide a great opportunity to process unstructured text to find meaningful relationships in literature.

Of particular relevance to this thesis, biomedical literature describes information about disease concepts as related to both humans and other organisms. One of the goals of the thesis was therefore to gather evidence from biomedical literature to further contribute to the assessment of potential disease organism models.

The methods developed throughout this work allow for the aggregation of data in different forms (e.g., graph and text) from a diverse set of biological data and knowledge bases to find meaningful patterns that are relevant to a given problem of interest. Graph-based models provide a conceptual framework to represent complex relationships in data (Campbell and Musen 1992, Lagoze et al. 2006, Zheleva and Getoor 2008). Knowledge-rich mathematical models for graph pattern analysis were developed and tested on diverse datasets with the aim to identify organism-dependent patterns. These models were tested on four genres of data: (1) genetic pathways, (2) chemical compounds, (3) molecular interaction networks, and (4) biomedical literature. Semantic hierarchies of terms such as Gene Ontology (Ashburner et al. 2000) and Medical Subject Headings (MeSH) (Lipscomb 2000) were used to annotate data to increase the generalization capability of the models and to address data sparsity issues. Using the methods developed in this thesis, evaluations were conducted to assess how a particular organism might be suited as a potential model for a particular class of diseases.

In the next section, a description is provided about the biological data and knowledge resources that were used in this study. An overview on related graph pattern mining methods is then presented. Concept graphs, as a means for graph-based representation for text, are then introduced. The chapter concludes with a summary of the major contributions of

this thesis and the subsequent chapters (which are based on accepted peer-review publications or on manuscripts in preparation) provide more details about each of the experiments conducted. The concluding chapter then presents possible future directions of this work.

1.2 Background

1.2.1 Biological Data Resources

Biological Interaction Networks

Biological interaction networks refer to undirected graphs that have nodes that represent entities in biological cells and links that represent interactions between these entities. Entities and interactions can be of many types, each of which define a network class. For instance, when physical interactions link molecular biology entities in a network, this network is referred to as a molecular interaction network. There are subclasses of molecular interaction networks depending on the types of nodes. For instance, when nodes represent proteins, the network is referred to as a protein-protein interaction network. Furthermore, if nodes represent regulatory proteins, transcription factors, and target genes, the resulting network is referred to as a gene regulatory network. Metabolic networks have nodes that represent enzymes and metabolites (chemical compounds). When interactions between entities are genetic (e.g., when function of a gene is affected by mutations in another gene), the network is referred to as a genetic interaction network.

Biological interaction networks (with physical and genetic interaction types) present a major data resource in cell biology, molecular biology and biomedical research generally.

Efforts have been made to study potential interactions between genes, proteins, and other molecules in cells (Cordell 2009, Marcotte et al. 1999, Sengupta et al. 1996). These interactions can be studied empirically (e.g., using yeast two-hybrid screening (Ito et al. 2001)) or can be predicted computationally (e.g., using comparative genomics or text mining techniques (Ferrer et al. 2011)). Freely accessible databases provide genetic/physical, experimental/predicted interactions for a wide range of model organisms. Some databases are manually curated (e.g., BioGRID (Stark et al. 2006)), which means no computational methods were used to predict interactions. Manually curated databases pose a data sparsity problem when dealing with concepts that are not widely studied (e.g., non-traditional model organisms) (Baumgartner et al. 2007). To fill this gap, methods for predicting interactions have been developed and used to build interaction networks for organisms with less supporting experimental data. These methods (e.g., as in the PIPs database (McDowall et al. 2009)) aggregate evidence from a variety of sources (e.g., those that catalogue gene co-expression or orthology [shared ancestry] information). For instance, the level of orthology between genes/proteins can be used to infer interactions in a potential model organism using pairs of interacting genes/proteins from a canonical model organism (Chautard et al. 2009). Text mining techniques can also be used to predict interacting pairs of genes/proteins (e.g., as in IntAct database (Kerrien et al. 2007)).

Genetic Pathways

Biological cells respond to a variety of events that take place inside and outside their perimeters. Chains of reactions within biological cells can start in response to certain events (e.g., upon detection of a signaling molecule coming from the outside of the

cell). Reactions inside biological cells can be described using conceptual models known as genetic pathways. A genetic pathway can be defined as ”a linear sequence of gene activities resulting from the functional interactions between different genes” (Faro et al. 2012). Genetic pathways are used to describe a wide range of mechanisms (e.g., biological functions (Walhout and Vidal 2001) or disease-related processes (Zhernakova et al. 2009)). In addition to genes, genetic pathways can contain other molecules such as gene products (e.g., proteins) and chemical substances (e.g., Calcium). Genetic pathways can be modeled as directed graphs with vertices representing molecules (e.g., genetic material, gene product, chemical molecule) and edges representing interaction types (e.g., inhibition, expression). In some cases, one vertex can represent a complex (a set of proteins that interact together as a unit).

Genetic pathways can be categorized according to the biological functions or mechanisms that they describe. For instance, metabolic pathways describe a set of chemical reactions related to metabolism inside cells (Schilling et al. 2000). Signaling pathways are used to describe interactions related to gene expression and cell communication (Li and Hristova 2006). Some disease mechanisms can be described using genetic pathways that include genes with mutations or gene expression levels hypothesized to be correlated with diseases (Vogelstein and Kinzler 2004). Genetic pathways that are related to diseases emphasize the importance of system-level representation of interacting genes to describe disease mechanisms by showing contexts (e.g., other reacting genes) of a mutant gene (Lin et al. 2007). Thus, research efforts have been devoted to discovering pathways that provide more understanding about cell mechanisms (Bomken et al. 2010). In addition to discovery of pathways, there have been efforts to discover missing genes in previously described

pathways (Osterman and Overbeek 2003). Discovery and analysis of disease pathways has gained importance in therapeutic research, because they provide system-level perspectives of complex diseases (in contrast to Mendelian disorders that are caused by single gene mutations) (Butcher et al. 2004). Finally, analysis of disease pathways may be useful to find candidate molecular targets for potential drugs (Hennessy et al. 2005).

Semantic Annotation of Biomedical Data

Ontologies, semantic networks, and controlled vocabularies are of great importance for the development and evaluation of computational methods in biomedical contexts (Shatkay and Feldman 2003). The mapping of raw data to terms or concepts in semantic hierarchies is a basic function that can be performed manually by domain experts (e.g., as in the case of determining the molecular function to a new gene sequence (Tweedie et al. 2009)) or by automated software tools (e.g., semantic annotation tools (Erdmann et al. 2000)). Data management methods for biological data can utilize ontologies and controlled vocabularies resources to perform tasks such as indexing and query processing. For instance, the PubMed interface for MEDLINE provides enhanced search capabilities when user queries are associated with applied controlled terms (MeSH descriptors) (Chang et al. 2006). In this thesis, terms from semantic hierarchies were used to annotate graph vertices that allowed for data abstraction and improved the generalization capability of the machine learning algorithms used. In particular, two semantic hierarchies were used for the studies described as part of this thesis: (1) Gene Ontology and (2) Unified Medical Language System (UMLS) Metathesaurus.

The Gene Ontology (GO) is a project that has developed and maintains a collection of ontologies, along with data processing tools for the applications of GO-based annotations of genomic materials. GO consists of three domain ontologies that organize concepts related to genes: (1) molecular function, (2) biological process, and (3) cellular component. The molecular function domain describes molecular level activities (e.g., binding or catalytic). The biological process domain describes processes (e.g., cell death) that may be accomplished by sequences of molecular functions. The cellular component domain describes locations (e.g., inner membrane) within cells. Using this set of ontologies, GO annotations have been assigned to gene products for numerous organisms, including those that serve as models for studying disease (Twigger et al. 2007). GO annotations can be determined based on experimental evidence and expert review of published literature (Conesa et al. 2005). Moreover, comparative genomics methods can be used to transfer annotations from data of one organism to newly sequenced genomic material of other organisms (Ferrer et al. 2011). The collection of GO annotations for a given organism are available through databases such as FlyBase (Fruit Fly)(Tweedie et al. 2009), RGD (Rat) (Dwinell et al. 2009), and MPact (Yeast)(Guldener et al. 2006).

The UMLS, developed at the United States National Library of Medicine (NLM), includes three biomedical knowledge resources (Bodenreider 2004): (1) Metathesaurus, (2) Semantic Network, and (3) the SPECIALIST natural language processing tools (Browne et al. 2003). Of relevance to this thesis, the Metathesaurus consists of concepts sources from over 100 biomedical hierarchies (including MeSH). Concepts from UMLS Metathesaurus can be used to annotate biomedical text publications using tools such as MetaMap, also developed at the NLM(Aronson 2001). MetaMap is available both as a stand-alone

application or as an Application Programming Interface (API) component. Advanced text processing features (e.g., Part-of-Speech (POS) tagging and syntactic parsing) are also available as part of MetaMap. The studies in this thesis leveraged MetaMap annotations of the MEDLINE corpus as available through the MetaMap Machine Output (MMO) - 2012 release. The MMO output files are available in text format with a defined structure. The MMO release contains a collection of individual files, each of which covers a range of citations referred to by unique PubMed Manuscript ID (PMID) identifiers. Each MMO output file contains several morpho-syntactic annotations including: part-of-speech tags for sentence words and phrase structures that determine the head of the phrase (e.g., important concept). In addition, MMO output files contain semantic annotations generated by MetaMap. In this thesis, the morph-syntactic and semantic annotations were used to construct concept graphs for mining biomedical literature.

1.2.2 Graph Pattern Mining Methods

Frequent Pattern Mining

A popular approach to feature extraction in graph data is the extraction of subgraphs (with defined quantitative criteria) that represent candidate features (Ranu and Singh 2009). Feature filtering techniques are commonly then used to select features whose distribution correlate with the distribution of graph class labels (Yu and Liu 2004, Fei and Huan 2008). A vector representation of subgraph features can be used by machine learning tools to solve learning problems (e.g., classification and clustering (Kong et al. 2011, Vogelstein et al. 2011)) or data management problems (e.g., graph indexing (Yan et al. 2004, Wang et al. 2012)). Searching for informative features can be achieved using feature interestingness measures. Examples of interestingness measures include mutual information, confidence,

support, and information gain. Tan, et al. have described 21 measures that can be used to identify and rank potentially interesting features (Tan et al. 2002).

Subgraph pattern mining methods have been used to address a range of graph data learning problems. In clustering problems, informative subgraph patterns can be used as features in vector space, and then clustering techniques can be applied to data represented in this feature space (Kulis et al. 2009). In classification problems, using subgraph patterns in feature vector representation of graphs can outperform graph embedding and kernel based methods in terms of accuracy and efficiency. For example, the LEAP search algorithm for significant pattern search combined with Support Vector Machines (SVM) can yield better results compared to kernel-based methods (Yan et al. 2008). By contrast, the gBoost classifier tightly combines a frequent subgraph pattern mining model with linear programming for graph classification and performs better than the frequent substructure mining approach because gBoost uses graph class labels in the search for subgraph features (Saigo et al. 2009). GraphSig is another approach that uses a scalable subgraph feature selection method to extract significant subgraphs using local patterns (i.e., paths) inside subgraphs captured by random walks (Ranu and Singh 2009). GAIA is an evolutionary computation algorithm for extraction of significant subgraphs in graph datasets and has been shown to perform well at the task of mining in chemical compounds (Jin et al. 2010).

Graph Kernels

In machine learning, kernel based methods (e.g., support vector machines [SVMs] and kernel principal component analysis [KPCA]) have been successfully applied to a range of learning problems involving various data types (e.g., text, graphs, and genome

sequences)(Hofmann et al. 2008). Graph kernels have been developed for the analysis of graph and network data to address structural pattern analysis and graph classification and clustering problems. Graph kernel methods have been applied in bioinformatics (e.g., analysis biological networks (Aittokallio and Schwikowski 2006) and protein function prediction (Borgwardt et al. 2005)), pattern recognition (e.g. image classification (Harchaoui and Bach 2007)), and chemical informatics (e.g., molecular fingerprinting (Ralaivola et al. 2005b)).

Graph kernel methods transform complex structured data (usually non-linearly separable) into a feature space where the data, in the transformed representation, can be separated approximately linearly (Hofmann et al. 2008). Then, the core computations of the graph kernel methods can be performed via operations of matrix algebra. A graph kernel is a function $k(x, y)$ that measures the similarity between two graph objects x and y. The similarity in this case can be based on common structural patterns (e.g., paths or walks) that two graph objects share. Common subgraph structures include important connectivity information (represented by the vertex adjacency lists), in addition to vertex labels. Previous research has focused on the development of graph kernel methods that are based on structural patterns that efficiently represent semantics embedded in graphs (Vishwanathan et al. 2010). Examples of graph kernels include pattern diffusion (Kondor and Lafferty 2002), graphlets (Shervashidze et al. 2009), subtrees (Shervashidze and Borgwardt 2009), and cyclic patterns (Horv $et\ al.2004$). More details on graph kernel models and algorithms can be found in (Vishwanathan et al. 2010).

The structural pattern analysis model that is presented in this thesis can be compared to a class of graph kernels known as Marginalized Graph Kernels in which the similarity between two graphs is calculated by taking into account the amount of labeled sequences of nodes or walks that the two graphs share (Kashima et al. 2003). The model can also be compared to graph kernels based on graphlets (subgraphs). A summary of popular graph kernel methods follows.

Random Walk Kernels

Random walk kernels (Borgwardt et al. 2006) construct a direct product graph for two input graphs. A direct product of two graphs G and H is a graph I for which the vertex set is the Cartesian product of the vertex sets of the input graphs (i.e., $V(G) \times V(H)$) and there exists an edge between two vertices $x = (a, b)$ and $y = (c, d)$ if and only if there is an edge between a and c in graph G and there is an edge between b and d in graph H. Then, every vertex in the direct product graph then represents a pair of nodes from the original input graphs. Random walking on a direct product graph I is the process of generating a vertex sequence such that every subsequent vertex is chosen according to the last vertex in the sequence and a transition probability function defined on the adjacency matrix. This probability function determines the next vertex to be picked given the identity of last vertex chosen so far. A random walk on the direct product on two graphs corresponds to a simultaneous random walk in the two original graphs. The number of walks in the direct product of two graphs quantifies the similarity between these two graphs.

Subtree Kernels and Cyclic Pattern Kernels

Graph kernels that are based on subtrees and cyclic patterns can capture more semantic

and structural information in a graph (compared to random walks). Subtree patterns within graphs can be identified by designating a vertex as the tree root and then adding all vertices that are reachable from root in a certain number of steps (these steps are called tree height) (Shervashidze and Borgwardt 2009). This specification can be implemented by a variety of techniques. For instance, a subtree graph kernel based on Weisfeiler-Lehman test of isomorphism is a fast method that can scale to large graphs (Shervashidze and Borgwardt 2009). Kernels based on cyclic patterns (Horv $et\ al.2004$) count the number of cycles shared by two graphs, limited to a predefined number of simple cycles, because computing general cycles is NP-hard.

Shortest-paths Kernels

Kernels based on shortest paths within graphs (Borgwardt and Kriegel 2005) have a computational advantage, since shortest paths can be identified in polynomial time while they express the inherent semantics in graphs. A first step toward computation of shortest-paths kernel is to transform original graphs into shortest-paths graphs using Floyds algorithm (Floyd 1962). Then, the shortest-paths kernel can be defined on edges of the Floyd transformed graphs.

Graphlet Kernels

Graphlets are subgraphs with a small number of vertices. For subgraphs of order k (i.e., of k vertices), vectors with components indicating frequency of a subgraph of order k are used to measure the similarity of two input graphs (Shervashidze et al. 2009). Each graph is represented by these count vectors and the kernel function is defined on this representation. Graphlet kernels have a scalability advantage to process datasets of large graphs. This

approach is similar to frequent pattern mining approaches to graph classification problem (e.g., GraphSig (Ranu and Singh 2009)) where feature vectors of graphs indicate presence or absence of significant subgraphs within a graph item in dataset.

1.2.3 Concept Graphs

The concept graphs (CGs) (Sowa 2008) formalism was defined by the linguist John F. Sowa as a basis for database support to process natural language queries. These graphs have two types of vertices: (1) concepts and (2) concept relations. A concept is an undefined primitive that can represent entities such as objects, actions and places. These entities do not have to be defined. Concept relations are used to connect concepts. Concept graphs are bipartite: i.e., edges cannot link vertices of the same type. A CG may consist of one vertex of a concept type. Concept relations vertices must have edges (cannot be isolated in the graph).

Concept graphs have been used for data representation when attempting to solve problems not necessarily related to database query processing. Natural language utterances can be perceived as a surface form (obeying linguistic rules of grammar and style) that embed interrelationship between concepts. As a formalism for graph-based data representation, CGs can be used to map natural language text utterances into a graph dataset. This mapping needs the syntactic relationships between words to be determined by assigning syntactic semantic labels to words in an utterance.

Construction of CGs has four steps. The first step is the annotation of natural language text to define its syntactic structure. Linguistic resources (e.g., lexicons, Part-of-Speech

(POS) taggers, and syntactic parsers) are necessary to compute syntactic structures of utterances. In the second step, the semantic roles of constituents of a given sentence are determined based on linguistic knowledge resources such as WordNet (Miller 1995) and VerbNet (Kipper et al. 2006). A semantic role represents what a noun phrase plays with respect to a verb in a given sentence. For instance, if a noun is determined as a subject of a transitive verb, then the semantic role of the noun is agent. Several semantic roles can be assigned to noun phrases including patient, recipient, and cause. The third step is to generate a subgraph for each semantic role defined in an utterance. Finally, a concept graph is generated by linking head vertices (i.e., main concepts) of subgraphs using concept relations determined by transformation grammars.

1.3 Contributions

The goal of the thesis was to develop computational methods that could exploit large, diverse, and high-dimensional datasets (e.g., as in graphs/networks and texts), in addition to available ontology-based annotations, to evaluate organisms as candidate disease models. The approach included developing mathematical models and algorithms to integrate knowledge sources and datasets in a way that can: (1) address the data sparsity problem, and (2) increase the generalization capability (e.g., learning relationships within one dataset so that these relationships can be useful in the analysis of other datasets).

The experiments carried out during this thesis were of four types. First, genetic pathways of human diseases were analyzed using a graph pattern mining algorithm to identify patterns that may correlate with a given disease type (e.g., cancer). Second, the

graph pattern mining method developed for this thesis was benchmarked using chemical compounds datasets, comparing the developed method to state-of-the-art graph kernels and frequent pattern mining methods. Third, the previously identified human disease patterns from genetic pathways were used to predict pathways in biological interaction networks for a number of organisms, aiming to rank each organism according to the degree to which its interaction network covers a given disease pattern. Finally, a fourth experiment evaluated the potential to identify disease model organisms from biomedical literature using a method that combined graph pattern mining with natural language processing techniques. What follows is a summary of each experiment, each of which are detailed in the subsequent chapters of this dissertation.

1.3.1 Analysis of Disease Patterns in Genetic Pathways

Current state-of-the-art models for graph pattern analysis and classification have some limitations regarding the processing of sparse and high-dimensional datasets of genetic pathways. Given that many genetic pathways may be unknown, it can be typical to have a small number of genetic pathways in a given graph dataset. Each graph may contain a set of vertices with a diverse set of labels (labels are drawn from a large space of possible genes, substrates, and proteins), it can be challenging to extract subgraph patterns that meet a given support value (e.g., frequency threshold). Genetic pathways that contain genes with known mutations to cause disorders are of interest in biomedicine. Disease pathways therefore offer a valuable resource to support a system-level study of complex diseases (e.g., diseases believed to be caused by genetic interactions). Analysis of graph datasets of disease pathways may highlight molecular mechanisms of diseases

at a system-level through identification of significant patterns. Disease pathways can be categorized according to disease types (e.g., immune versus infectious diseases). To test the hypothesis that functional structural patterns of interacting biological entities (e.g., genes) correlate with disease classes, a graph classifier was developed for genetic pathways.

Approach

The developed graph pattern analysis method leveraged Gene Ontology (GO) annotations to label graph vertices to indicate general molecular functions of genes/proteins. This vertex labeling scheme addressed the data sparseness problem by mapping a set of genes/proteins that shared the same molecular function to one label value. The labeling scheme had three benefits. First, it increased the frequency of vertex labels in a given graph dataset. Second, subgraph patterns were perceived as molecular functional patterns within biological processes that enabled the analysis of genetic pathways at a functional level. Third, functional subgraph patterns enabled a more generalized capability to allow for the matching of identified patterns to other genetic pathways.

A statistical model was developed to provide a quantitative measure for evaluating subgraph patterns. A mathematical function, termed graph partitioning, was defined based on edge sets of a graph and used to map a given graph onto a set of edge-disjoint subgraphs, each of which were assumed to represent a feature. Typically, there is a large space of possible partitionings. The statistical model provided a means to score partitionings. The Expectation-Maximization (EM) algorithm was used to estimate parameters of the statistical model while searching for the most likely partitionings in a graph dataset. The

set of best partitionings highlighted key functional subgraph patterns, termed disease fingerprints, in disease pathways.

For the pathways classification task, a naïve Bayes graph classifier was developed based on the graph pattern analysis model. Training data were analyzed to identify the best graph partitionings and estimate model parameters. Then, the graph classifier was used to analyze test data to search for most likely partitionings using model parameters estimated during training. This naïve Bayes classifier allowed for incorporation of external knowledge source such as a priori distribution of graph class labels. This enabled the graph classifier to process unbalanced datasets (i.e., when the distribution of class labels is highly skewed). The classifier was tested on genetic disease pathway datasets from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database.

Contributions

The contributions of this study to this thesis are a method that:

- Addresses the data sparseness problem of disease pathway graph datasets through annotation of genes/proteins with molecular function
- Provides a statistical model for graph pattern analysis
- Utilizes a search algorithm to explore large space of subgraph patterns
- Identifies disease fingerprints from disease genetic pathways
- Leverages a naïve Bayes' classifier for graphs that addressed the problem of genetic pathway classification

Major Results

The developed graph pattern analysis and classification model was evaluated on a dataset of 56 disease genetic pathways from the KEGG database. The analysis revealed that disease fingerprints in genetic pathways of cancer disease and infectious diseases could be identified with graph partitionings of disease pathways. High scoring fingerprints were extracted and made available for domain expert review. Based on a cross-validation of the KEGG pathways dataset, the method achieved a Positive Predictive Value (Precision) of 0.77 for cancer type pathways and of 0.6 for infectious type pathways, as well as a Sensitivity (Recall) of 0.83 and 0.75 for cancer type pathways and infectious type pathways, respectively.

1.3.2 Graph Pattern Analysis in Drug Chemical Compounds

Classification of chemical substances and compounds according to chemical activity in anti-cancer screen experiments is a challenging problem. Given the large numbers of possible chemical compounds (tens of thousands), there is a need for automated tools for prediction of chemical compounds activities. Chemical compounds can be modeled as indirected graphs with vertices representing atoms and edges representing bonds linking atoms. Unlike biological interaction networks that have a large number of genes/protein names, the distribution of vertex labels in chemical compounds is highly skewed, with four or five elements (including Hydrogen, Carbon, Nitrogen, and Oxygen) representing more than 90 percent of vertex labels in a chemical compound graph dataset. This can be a challenging factor for machine learning algorithms to accommodate. For instance, a chemical compound of 12 atoms may have only two unique labels (e.g., Benzene [C6H6]). This necessitates the search for structural pattern features other than single vertex labels. These

complex structural features are based on graph connectivity and can be represented by microstructures such as walks (maximal paths or cycles) and trees. Some graph pattern analysis models utilize cycles within graphs (e.g., cyclic pattern graph kernels (Horv $et\ al.2004$)). However, patterns with very similar vertex labels are of low discriminative power for a task such as clustering and classification. For instance, many organic compounds consist of rings of five or six carbons (e.g., Cyclopentane [C5H10] and Benzene [C6H6]). This characteristic would make rings of carbon atoms of little discriminative power for task such as classification.

Approach

A graph pattern analysis model was developed that put greater emphasis on graph connectivity while searching for key subgraph patterns (e.g., trees, cycles, and any general pattern with a subset of edges). This model provided a quantitative measure of feature quality that was used by a heuristic search algorithm to explore a large space of subgraph patterns and to identify discriminative features within items of a graph dataset. Walks were essential for capturing graph semantics defined by the vertex connectivity and the function of heuristic search algorithm required keeping track of walks inside subgraph patterns. A subgraph pattern was then approximated by a set of walks, and the probability of observing a subgraph pattern in a dataset was approximated as a function of probabilities of included walks. This method may have a good generalization capability by being able to compute the probability value of observing a new subgraph pattern (not in the training dataset) and making use of the fact it might share many walks with other pattern previously found in the training data. A second advantage of the proposed approach is that approximating a subgraph by

walks avoids the computational overhead of testing whether one subgraph is isomorphic to another.

Contributions

The contributions of this study to this thesis was a new method that:

- Estimated subgraph pattern frequency by approximating patterns by sets of walks
- Was empirically evaluated on seven datasets and compared to two state-of-the-art methods of graph pattern analysis: graph kernels and frequent pattern mining
- Included the building of two graph classifier systems: (1) a naïve Bayes classifier and (2) a Support Vector Machines (SVM) classifier.

Major Results

The developed graph pattern analysis method was tested in the context of a graph classification task for seven chemical compounds datasets. The performance was compared to four graph kernel methods and one frequent pattern mining method. In addition to classification accuracy, a t-test was used to ascertain if the best performing classifier for a given analysis was significantly better than others. The frequent pattern mining classifier did not achieve significant accuracy. Graph kernels provided significant performance on three datasets, while the method developed for this thesis provided significant accuracy on three other datasets. For one dataset there was no clear winner. It is important to note that two out of four graph kernel methods were not able to complete computation for two out of seven datasets. For these two datasets, the system developed for this thesis was able to com-

plete its computation, indicating a computational tractability advantage for the developed method.

1.3.3 Assessment of Organisms as Potential Disease Models

Model organisms are of great importance in biomedicine, providing systematic and controlled environments to study and uncover the underlying mechanisms of diseases. Mice (Mus musculus) and rats (Rattus norvegicus) are dominantly used as model organisms; however, there may be other organisms that could be used as models for certain diseases. For instance, McGary, et al. suggested a yeast model (Saccharomyces cerevisiae) for angiogenesis disorders, a worm model (Caenorhabditis elegans) for breast cancer, a plant model (Arabidopsis thaliana) for Waardenburg syndrome, and a mouse model for autism (McGary et al. 2010). As the McGary, et al. study demonstrates, high throughput sequencing techniques provide more genetic material data for organisms than ever before and recent advancements in comparative genomics allow for prediction of putative gene functions and also for predicting molecular interactions between pairs of genes. Thus, there may be an opportunity to leverage automated approaches to provide *in silico* prediction of organisms as potential disease models.

Approach

A method for automated analysis of biological interaction networks and human disease genetic pathways was developed to assess organisms as potential disease models. The approach leveraged knowledge about human diseases as present in genetic pathways, organism biological interaction networks, and gene molecular function annotations using the Gene Ontology to rank potential model organisms for a given disease category. The ap-

proach starts with learning of characteristic functional patterns of diseases within genetic pathways. Then, the characteristic functional patterns were matched to subnetworks in biological interaction networks. These subnetworks were hypothesized to be parts of (potential new) genetic pathways and were compared to a set of reference disease pathways. The quality of a candidate model organism was then determined by the degree to which its biological interaction network covered patterns that were found to match reference disease pathways.

Contributions

The contributions of this study to the thesis are:

- A molecular and graph-based methodology for evaluation of potential model organisms
- A graph indexing and query processing method to allow efficient matching of query subgraph patterns to large interaction networks
- Prediction of genetic pathways in biological interaction networks

Major Results

Two major findings were reported for this method. The first major finding pertained to the effectiveness of using knowledge-based models of graph pattern analysis for increasing the generalization capability of patterns. Generalized patterns had nodes with two types of annotations: (1) gene/protein name and (2) molecular function. Using generalized patterns that were identified within genetic pathways, the method was able to recover a number of known pathways (already published in databases of organisms), with the potential of

discovering unpublished pathways. The study revealed that one generalized pattern can be successfully matched to subnetworks within biological networks of different organisms, and these subnetworks share same structure (links) and molecular annotation of nodes, but sets of gene/protein names in nodes might be different. A major finding was related to the effectiveness of using predicted interactions between genes to increase the size of interaction networks to cover more patterns. Some interaction networks can contain hundreds of manually curated interactions. These small-sized networks cannot provide the necessary coverage of generalized patterns and therefore it was necessary to utilize predicted interactions (using comparative genomics and text mining methods) to increase network sizes.

GO-annotated subgraph patterns (fingerprints) within genetic pathways of diseases were matched to GO-annotated biological interaction networks of 14 organisms. The assessment of each organism was based on coverage of biological networks to disease fingerprints. Pattern coverage was calculated as the percentage of subgraph patterns that successfully matched a subnetwork in an organism's network such that the gene/protein names within this subnetwork were found in a reference (known/published) pathway. For each disease, organisms were ranked based on performance in terms of pattern coverage. A number of organisms (besides mice or rats) were found to be highly ranked with regard to coverage of disease patterns. For instance, the plant *Arabidopsis thaliana* (mouse-ear cress) and bacterium *Escherichia coli* were found to be the best possible model organisms for colorectal cancer and thyroid cancer, *Sacchromyces cerevisiae* (Bakers yeast) was found to be a possible good model for *Eppstein-Barr* virus disease, and *Danio rerio* (zebrafish) was found to be a possible good model organism for Renal cell carcinoma,

Melanoma, and Pertussis.

The second major finding of this method was the empirical evaluation of the contribution of predicted interactions to successful prediction of new pathways. Some organisms had few interactions with evidence that was experimentally supported. Using predicted interactions inferred by genome-context methods such as gene fusion and gene-neighborhood methods helped increase network size and hence increase the chance of a generalized pattern to have match to a subnetwork. For instance, there were about 112 interactions for *Danio rerio* (Zebrafish) in the manually curated database BioGRID. This network is very small to cover patterns of genetic pathways and thus decreased the chance of predicting new pathways. With predicted interactions imported from the meta-database STRING, 47,000 interactions were added to the network for *Danio rerio*. This had a positive impact on performance of this organism as a potential model organism in terms of successfully predicting new pathways.

1.3.4 Graph-based Mining in Biomedical Literature for Assessment of Disease Model Organisms

One of the goals of biomedical research efforts is to discover genes and their functions in addition to the related molecular mechanisms (or pathways) underlying cellular processes in organisms. Methods of comparative genomics have been developed to predict gene functions using information about similar genes of previously studied organisms. This has led to an increasing number of research reports that describe newly discovered mechanisms underlying biological phenomena, in particular, those related to disease etiology. Scientific knowledge may be represented by relationships between domain scientific concepts men-

tioned in literature. However, this knowledge cannot be easily accessed and summarized to address various scientific questions, because it is generally embedded in free text. In order to discover patterns in text data, raw text content need to be annotated with morpho-syntactic and semantic information that describe the structures of sentences. Data mining methods, coupled with natural language processing (NLP) methods, may thus provide a means to identify significant patterns that summarize information embedded in biomedical literature. The availability of biomedical knowledge in high-volume, freely available resources (e.g., as citations indexed by MEDLINE) provides an opportunity to leverage data mining methods to test data-driven hypothesis about biological phenomena.

Approach

A method was developed to generate graph-based representations of text sentences using available morpho-syntactic information as well as semantic annotations. This graph-based representation maintains a labeling scheme in which each vertex has multiple annotation types, termed factors. A statistical pattern analysis model was developed to help identification of significant patterns (termed fingerprints) in graphs. This pattern analysis model combines annotation factors information stored in each vertex of the graph. Towards the goal of assessment of potential model organisms, the methods used NLP annotations of biomedical citations (MEDLINE) to generate a graph-based representation of sentences in biomedical citations (MEDLINE), and then applied the graph pattern analysis model to uncover significant subgraph patterns. The pattern model allowed for the incorporation of ontology-based annotations from knowledge bases to address the data sparsity problem (that is particularly common with textual data) and to increase the generalization capability of subgraph patterns. Generalization of subgraph patterns in this context meant that

vertices in patterns were annotated with terms from ontologies that enabled patterns to be matched to other patterns within graphs whose vertices were also annotated with the same ontological terms.

Contributions

The contributions of this study to the thesis are:

- A new graph-based method for text pattern analysis allowed for application of graph pattern analysis model to uncover complex relationships in text
- A multi-factor vertex annotation scheme and a factored graph pattern analysis model enabled utilization of rich annotations of text data to address the data sparsity problem that is common with textual data
- The method was applied to a significant problem, which is how to summarize content of biomedical citations using statistically significant patterns. This method enabled the analysis of large number of citations to extract patterns about biological phenomena. These patterns were used to assess organisms as potential disease models

Major Results

The proposed methods allowed for the processing of a corpus of nine million sentences that represents biomedical abstracts related to organisms and biological processes and phenomena. A total of 82 organisms were evaluated as potential models for six disease categories. For each organism, a set of significant patterns summarized evidence on how this organism can serve as a disease model for particular disease categories. A total of six significant pattern sets were generated for each organism. Each of the pattern sets of an

organism was compared to the corresponding pattern set of humans. The proportion of matching between an organisms pattern set and the humans pattern set gave an indication on whether this organism can serve as a better disease model.

The experiments suggested that there are potentially good candidate organisms for some diseases in addition to the widely used organisms in laboratories. For instance, in the case of cardiovascular diseases, Oncorhynchus mykiss (trout) performed the best as a potential model organism, along with Danio rerio (Zerbafish) and Drosophila melanogaster (fruit fly). In addition, the Torpedo torpedo (Torpedo fish) and Gallus gallus (chicken) had the best matches as potential models for immune system diseases. For nervous system diseases, birds (especially G. gallus) had the best matches to human fingerprints. D. rerio still performed the best as a potential model for Endocrine system diseases.

1.4 Thesis Organization

Chapter 2 through 5 are based on manuscripts that have been published or are in the process of being peer-reviewed. In Chapter 2, the problem of mining disease patterns within genetic pathways is formally defined. A mathematical model for pattern analysis and an algorithm for parameter estimation are presented. In addition to finding disease patterns, the problem of classification of genetic pathways is addressed using a naïve Bayes classifier based on the developed model. Chapter 3 presents a benchmarking of the developed graph pattern analysis method through analysis of chemical compounds datasets (which are larger in size, structurally different, and computationally harder to process than genetic pathways). Performance of the developed method was compared to two different graph analysis methods (graph kernels and frequent pattern analysis). Chapter 4 presents the

problem of evaluation of organisms as disease models using biological interaction networks and knowledge resources. A graph indexing and query processing method was developed to provide knowledge-rich access to data. In Chapter 5, the problem of evaluation of model organisms was addressed using an approach that combined natural language processing with graph pattern mining through graph-based analysis of biomedical literature. Chapter 6 presents over-arching conclusions and presents a discussion on potential future work.

Chapter 2

Mining Disease Fingerprints From Within Genetic Pathways

Nabhan, A. R. and I. N. Sarkar (2012). Mining disease fingerprints from within genetic pathways. In AMIA *Annual Symposium Proceedings, Volume 2012, pp. 1320. American Medical Informatics Association.*

2.1 Abstract

Mining biological networks can be an effective means to uncover system level knowledge out of micro level associations, such as encapsulated in genetic pathways. Analysis of human disease genetic pathways can lead to the identification of major mechanisms that may underlie disorders at an abstract functional level. The focus of this study was to develop an approach for structural pattern analysis and classification of genetic pathways of diseases. A probabilistic model was developed to capture characteristic components (fingerprints) of functionally annotated pathways. A probability estimation procedure of this model

searched for fingerprints in each disease pathway while improving probability estimates of model parameters. The approach was evaluated on data from the Kyoto Encyclopedia of Genes and Genomes (consisting of 56 pathways across seven disease categories). Based on the achieved average classification accuracy of up to 77%, the findings suggest that these fingerprints may be used for classification and discovery of genetic pathways.

2.2 Introduction

Biological cells have sophisticated information processing systems with highly modular architectures. The flow of information in and between cells can be achieved through a series of biochemical interactions that are composed of a network with a fixed or changing topology. Gaining insight into the operations of cells requires the analysis of components (e.g., genetic material, chemical molecules, and compounds), identifying links (wiring) that represent relations or interactions between components, and discovering information pathways in these networks. Analysis of the structure and dynamics of biological networks plays an important role in understanding architecture and function of biological systems. To level the landscape for a system-based understanding of cellular processes, there has been much previous work in the construction of biological network models, accompanying databases, and development of identification (prediction) algorithms of genetic pathways (Rual et al. 2005, Kanehisa et al. 2010, Stelzl et al. 2005, Franke et al. 2006, Caspi et al. 2008).

Network medicine (Barabasi et al. 2011) represents one application area where the analysis of biological networks has a potentially direct impact on human health. In this regard, the analysis of genetic pathways may advance knowledge towards an understanding

of the molecular underpinnings of the disease process (Rudy et al. 2008, Karnovsky et al. 2012, Novoyatleva et al. 2010, Liu and Olson 2010, Slattery et al. 2009, Cogswell et al. 2008). Important questions about complex diseases, such as Alzheimer Disease and Parkinson Disease, have been explored by investigating genetic pathways (Lambert et al. 2010, Pan et al. 2008). Genetic pathways can also play an important role in drug discovery. For example, targeting a specific step in a disease pathway with the aim of identifying highly specific inhibitors can be used in drug development efforts (Pawson and Linding 2008). Additionally, pathway analysis has also been shown to be useful for analyzing groups of proteins in signaling or metabolic pathways with known functions to find more effective drug targets (Arrell and Terzic 2010).

Functional pathway analysis can be broadly classified into over-representation analysis (ORA), functional class scoring (FCS), or Pathway Topology (PT)-Based approaches (Khatri et al. 2012). In contrast to ORA or FCS, PT analysis takes into consideration structural and topological information about pathways, such as positions of genes in the pathway diagram, types of reactions, and number of reactions. This approach can be supported by knowledge within knowledge bases such as the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al. 2010), MetaCyc(Karp et al. 2002), and Reactome (Joshi-Tope et al. 2005). A potentially insightful aspect of pathway analysis includes the study of structural patterns that might be embedded within directed graphs. Studying such structural patterns could be used to identify major sub-processes that may be associated with major biological functions (e.g., regulation).

CHAPTER 2. MINING DISEASE FINGERPRINTS FROM WITHIN GENETIC PATHWAYS

The structural analysis of genetic pathways lies at the intersection of biomedical informatics, graph theory, and data mining (Maudsley et al. 2011, Huang et al. 2011, You et al. 2009). Many research efforts have been directed to the prediction and identification of pathway features of potential interest. You, et al. used graph substucture analysis to find biologically meaningful substructures in KEGGs metabolic pathways (You et al. 2009). Cakmak and Ozsoyoglu showed that functionality patterns in metabolic networks enriched with functional annotation of enzymes could be used to discover unknown pathways in organisms (Cakmak and Ozsoyoglu 2007). Battle, et al. used quantitative genetic interaction measurements within a Bayesian learning framework to identify pathways (Battle et al. 2010). Cerami, et al. combined an analysis of sequence mutations with a network analysis of molecular interaction networks to identify core disease pathways in Glioblastoma (Cerami et al. 2010). Chen, et al. used topological information of graphs to find optimal set of features to answer the question whether a module of proteins forms a meaningful pathway (Chen et al. 2010). Huang, et al. used feature set including graph properties, biochemical and physicochemical properties for pathway classification (Huang et al. 2011). Many of pathway analysis studies combine graph structure information, knowledge about genes and proteins at functional and biochemical levels.

The focus of this study was on the structural pattern analysis of genetic pathways of diseases. The particular goal of the study was to identify major components that may characterize disease classes, focusing primarily on complex disorders (i.e., disorders that involve multiple genes). For each disease category, distinctive functional and structural characteristics (fingerprints) were identified based on the training of a classification model using genetic pathways dataset.

2.3 Methods

The overall goal of this study was to develop an approach to identify unique characteristics (fingerprints) associated with a given disease class. The process started by annotating elements within a training set of disease pathways with functional annotations. These functionally annotated pathway graphs were then structurally analyzed to learn a probability model that accounted for both the graph structure and functional annotations. This model was used in pathway classification to assess the effectiveness of learning disease characteristics.

2.3.1 Functional Annotation of the KEGG Pathways Dataset

KEGG pathways are stored in files formatted according to the KEGG Markup Language (KGML), used to model genetic pathways. The KGML files were parsed using the BioRuby API (Goto et al. 2010) to extract nodes and edges that composed a directed graph. Edges were annotated in the KGML files with relation labels such as activation, phosphorylation, and expression. Nodes that represented genes were further annotated with functional annotations using the Gene Ontology (GO), based on information extracted from the Human Protein Reference Database (HPRD) (Prasad et al. 2009).

Each node in a KGML file can represent more than one gene. Furthermore, each gene may match more than one GO term in HPRD annotated dataset. Thus, there can be a list of GO terms for each entry in a given pathway. Because the proposed model for classification can handle only one annotation per node or edge, a preprocessing step was developed that

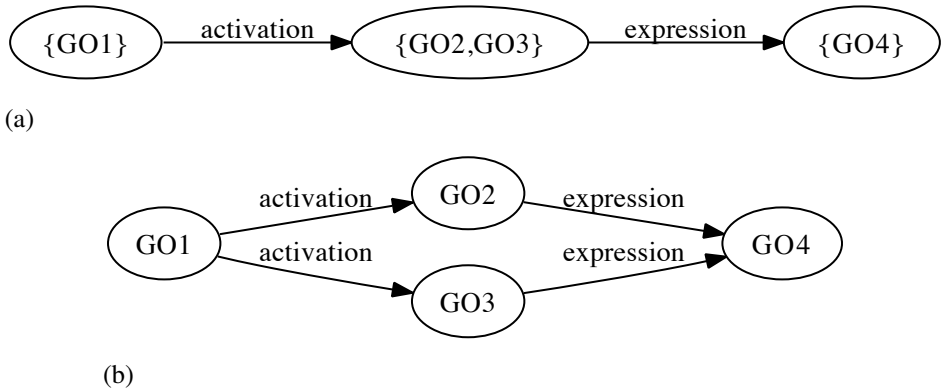


Figure 2.1: Node and edge replication. A node that has more than one GO annotation in graph (a) has been replicated in graph (b). As a consequence, edges have also been replicated in (b).

took nodes with more than one GO term and replicated them. Furthermore, each replicated instance of a given node carried only one GO term. Whenever a node was replicated, its incoming and outgoing edges were copied to link replicated nodes to their predecessor and successor nodes. Figure 2.1 illustrates this process.

2.3.2 Graph Representation of Genetic Pathways

Disease pathways were modeled as labeled directed graphs where nodes represented genes and edges represented relationships between genes. An example of a labeled graph is shown in Figure 2.2. Node V3 has a label F3 and Node V4 has label F4. An edge (E4) connecting this pair of nodes has the label activation. In addition to labeled nodes and edges, each disease pathway was associated in KEGG with a class label categorizing the nature of the disease. Examples of pathway class labels in the dataset include: cancer, infectious, and immune.

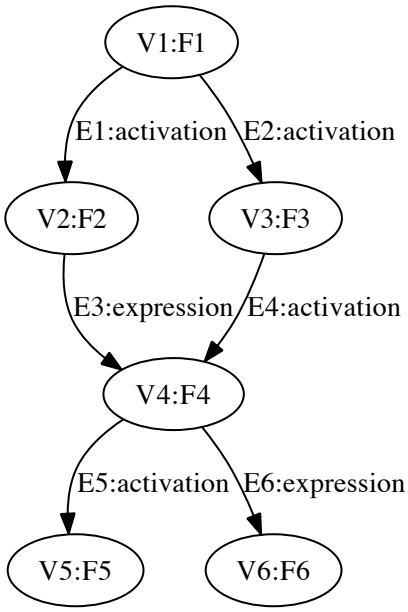


Figure 2.2: A labeled directed graph that represents a functionally annotated genetic pathway.

2.3.3 Mathematical Model

A particular class of diseases was assumed to have specific characteristics that make it distinct from other disease classes. The implemented model thus took into account associations between a particular disease class and pathways structure and annotations. Every graph instance, G, was considered as one of many possible examples that contained characteristics of a disease class C. Every pair of disease class and graph (C, G) was assigned a probability value $P(G|C)$, which was interpreted as a quantification of the amount of characteristics of disease class C contained in graph G. The system then aimed to find disease class C, given an observed graph G. These relationships can be expressed using Bayes theorem:

$$P(C|G) = \frac{P(C)P(G|C)}{P(G)} \tag{2.1}$$

Then, the goal of the classifier is to search for a disease class $\hat{C}$ for which $P(C|G)$ was the greatest, where

$$\hat{C} = argmax_C P(C)P(G|C) \tag{2.2}$$

This makes the assumption that the denominator of Eq. 2.1 was independent of C, thus suggesting that finding $\hat{C}$ was the same as finding C so that the quantity $P(C)P(G|C)$ was as large as possible.

Incorporation of Graph Substructures

The calculation of probability value $P(G|C)$ needed to take into account the possible structural patterns of G that could be considered characteristics of disease class C. A given graph can be decomposed in many ways into subgraphs, each of which can be considered a candidate characteristic of a disease class. The decomposition of a graph into its subgraphs was defined using the following definition for graph partitioning:

Definition 2.1 *A partitioning Φ of graph G is a function $\Phi : E(G) \rightarrow N$, where E(G) is the edge set of G and N is the set of natural numbers. A subset of edges {e1, e2 ... ek} is said to be in the same subgraph if and only if $\Phi(e1) = \Phi(e2) ... = \Phi(ek)$.*

A partitioning of a given graph is a set of subgraphs that are edge-disjoint (i.e., an edge belongs only to one subgraph). This partitioning can be represented by an array of integers where positions points to edges and content indicate a subgraph to which the edge in position belongs. To illustrate this definition, consider the following example.

Let $E(G) =< E_1, E_2, E_3, E_4, E_5, E_6 >$ be an ordered sequence of edges in a graph G. A partitioning can be represented as an integer array of length equal to the $|E(G)|$. A set of subgraphs S is created according to this partitioning. For each subgraph $g_i \in S$, edge set of g_i is $E(g_i) = \{e|\Phi(e) = i\}$. An example of a partitioning Φ is the sequence $< 1, 2, 1, 2, 3, 3 >$, which means that G can be divided into three subgraphs: g_1 containing edges E_1, E_3, g_2 containing the edge E_2, E_4, and g_3 containing the edges E_5, E_6. Figure 2.3 shows an example of this kind of partitioning.

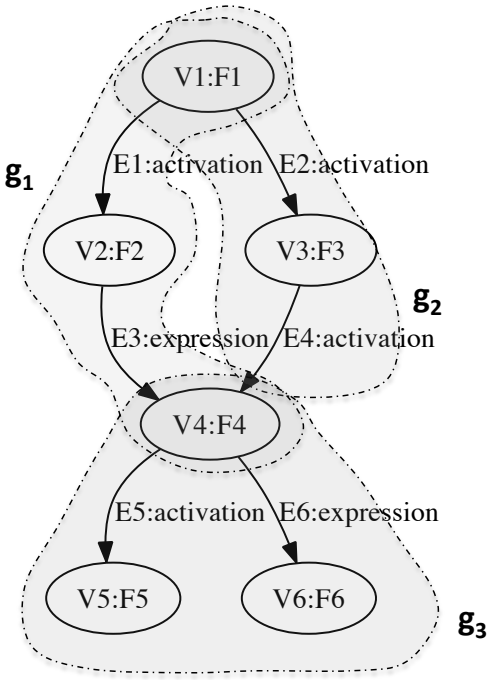


Figure 2.3: A partitioning of graph G into three subgraphs g1, g2 and g3.

To incorporate structural patterns in the calculation of $P(G|C)$, graph partitioning can be introduced as a hidden variable Φ, and hence $P(G|C)$ can be expanded as:

$$P(G|C) = \sum_{\Phi} P(\Phi, G|C) \tag{2.3}$$

A partitioning function naturally divides a graph into a set of features that can be used for classification. Since there might be many possible partitionings (some of them might be equally probable), a sum over partitionings is used in the right hand side of Eq. 2.3. Let $S = \{g_1, g_2...g_n\}$ be the set of subgraphs of G according to a partitioning Φ. The likelihood of an arbitrary partitioning Φ of graph G given a class C can be expressed as:

$$P(\Phi, G|C) = \prod_{g \in S} P(g|C) \tag{2.4}$$

Where, S is the set of non-overlapping subgraphs of graph G according to partitioning Φ: $S = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \Phi(e) = i\}$.

The product of $P(g|C)$ terms used in Eq. 2.4 assumes that subgraphs or features are orthogonal. The value $P(g|C)$ represents the likelihood that g is a characteristic of class C. To simplify calculation of $P(g|C)$, a subgraph g can be approximated by a set of maximal paths, A, inside g. Hence,

$$P(g|C) \approx \prod_{A \in g} P(A|C) \tag{2.5}$$

And then, one can combine Equations 2.4 and 2.5:

$$P(\Phi, G|C) = \prod_{g \in S} \prod_{A \in g} P(A|C) \tag{2.6}$$

Where, $A \in g$ denotes a maximal path A inside subgraph g. Finally, the probability of a given partitioning Φ can be calculated using

$$P(\Phi|G,C) = \frac{P(\Phi,G|C)}{\sum_j P(\Phi_j,G|C)} \tag{2.7}$$

Equations 2.3-2.7 suggest an approach to compute the conditional probability $P(G|C)$ in a tractable manner. To compute this probability, it was first required to generate partitionings. Then, the likelihood of each partitioning was calculated based on a conditional probability distribution of paths $P(A|C)$. Thus, finding paths inside subgraphs was needed to build and update $P(A|C)$. Therefore, the training procedure was based on finding paths inside subgraphs and utilizing the concept of partitioning to compute $P(G|C)$ according to Equation 2.3.

2.3.4 Model Training

The objective of probability estimation is to build the conditional distribution $P(A|C)$. This involves counting co-occurrence of pairs of path A and a class C. Since, for any given graph, there can be many different ways to decompose it into subgraphs, a single path may simultaneously belong to more than one possible subgraph. The question is how to count the co-occurrence of the path-class pair in this case? A possible solution is to weigh each occurrence of path-class pair by the probability of the partitioning Φ to which that path belongs. This step is called collecting fraction counts, since each occurrence of path-class pair is discounted by the probability of partitioning Φ. The idea of collecting fraction counts has been successfully applied to machine learning problems such as statistical machine translation (Brown et al. 1993).

Counting Class-Path Co-occurrences

To collect counts of path-class pairs, the probability of possible partitionings needs to be computed. In turn, computing probability of partitionings needs the conditional probability $P(A|C)$, which depends on counting co- occurrences of path-class pairs. This problem can be solved by an iterative training procedure using the Expectation Maximization (EM) algorithm (Dempster et al. 1977). The first step is seeding the partitionings: to generate a number of random partitionings (maximum number of partitionings is an adjustable parameter of the tool) for each graph. Instances of path-class pairs, (A, C), are then identified within each subgraph according to each partitioning. The counts of (A, C) pairs are used to create the conditional probability distribution $P(A|C)$. Thus, this iterative process has two phases: (1) E-Step: search for and compute the likelihood of each partitioning using Eqs. 2.6 and 2.7; and (2) M-Step: fraction counts of (A, C) pairs are collected, and better estimates of conditional probability $P(A|C)$ is produced. The number of training iterations is an adjustable parameter. Since the used graph dataset was limited in the number of items, it was not possible to reserve a portion of the dataset for parameter tuning. The number of EM iterations was thus adjusted empirically and the experiments on a randomized version of the dataset have shown that three iterations of EM training yields best results. The outline of this process is shown in Table 2.1 presents.

At the beginning of model training, the conditional probability table $P(A|C)$ is initialized with single-edge paths. There is a minimum probability value $\epsilon = P(A|C)$ for paths that are not discovered yet in early iterations of EM algorithm. In the E-Step, new paths are likely to be discovered when new partitionings (and probably larger subgraphs having longer paths) are explored in Hill-Climbing search for partitionings. These newly discovered paths are added to the conditional probability $P(A|C)$ when collecting fraction

Table 2.1: $P(A|C)$ probability estimation algorithm

Input:
D: graph data set $G_1, ..., G_n$
N: Number of iterations
Process
1: Create seed partitionings and Initialize $P(A|C)$ table with uniform probability value.
2: for $i = 1 : N$
E-Step
3: for each $G \in D$
4: Let C be the class label of G and let the set of graph partitionings $G.\Phi s$ = searchForPartitionings(G,C)
5: Use Eq. 2.6-2.7 to compute the likelihood of every partitioning $\Phi \in G.\Phi s$
M-Step
6: for each $G \in D$
7: for each $\Phi \in G.\Phi s$
8: for each $g \in S = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \Phi(e) = i\}$
9: for each maximal path $A \in g$
10: $CountTable(A, C)$+=$\Phi.probability$
11: Normalize entries of $CountTable(A, C)$ to obtain $P(A|C)$
Output: updated $P(A|C)$, $G.\Phi*$ //return conditional probability and best partitioning

counts in the M-Step.

2.3.5 Predicting Class Labels

Given a conditional probability model $P(A|C)$ for paths and class labels as well as a prior probability distribution model $P(C)$ for class labels ($P(C)$ can be computed using frequency of each disease class in the dataset), a new graph instance was classified as follows. A search for best partitioning for the target graph was performed, using all

possible candidate class labels. The evaluation of partitioning quality was measured using $P(A|C)$ according to Eqs. 2.4-2.7. The Hill-climbing search for set of best partitionings was performed. The candidate class label that maximizes Eq.2.2 was reported as target class label. Table 2.2 shows how classification was performed.

Table 2.2: Predicting a class label for a test graph instance

Input: Graph G, set of class labels C, paths conditional probability distribution $P(A\|C)$ and prior class probability distribution $P(C)$
Process
1: For each class label $\ell \in C$
2: Using the probability distribution $P(A\|C)$, Φ_s = searchForPartitionings(G,ℓ)
3: Compute $P(\ell)P(G\|\ell)$ according to Eqs. 2.3-2.6 using the set of best partitionings
Output: Class label with the highest $P(\ell)P(G\|\ell)$ value

Extracting Disease Fingerprints

Fingerprints were defined as subgraphs representing structural patterns and were extracted from the best partitionings of graph instances. These sets of subgraphs were considered key characteristics of disease classes and highlight major processes (e.g., chains of reactions) inside a disease pathway. Fingerprints were extracted from the best partitionings that had probability scores greater than a specified threshold value ($\delta > 0.1$), which represents confidence about partitioning quality. The choice of the threshold value depends on the size of graphs (in terms of edges) and the number of graphs in the dataset. This threshold helps one to choose only highly probable partitionings for manual inspection. High threshold values tend to print low numbers of partitionings to disk files. If more partitionings need to be examined, a slightly lower threshold value can be used. To show the importance of structural patterns in identifying macro-level view of each disease pathway, maps were generated to

represent joint distributions of GO terms. The intensity in these maps reflected how often two GO terms were linked together by an edge in the graph. Since edges can have annotations for a set of basic processes such as expression or phosphorylation, a map was generated for each process type. Thus, maps were generated for expression, phosphorylation, activation, etc. The spatial patterns of these maps enabled a global view of GO terms connectivity within the complete data set. Nodes of subgraph fingerprints were mapped onto points in maps to see if nodes of subgraph fingerprints tended to cohere (found to be near each other) in the map. The maps were developed to highlight the utility of structural patterns in contrast with micro-level patterns that emerge from graph-theoretic properties such as edge degree distribution.

2.3.6 Experimental Settings

Datasets

Pathway diagrams were downloaded from KEGG Pathway database (December 2011). GO annotations were extracted from the Gene Ontology file of HPRD. This dataset was composed of 56 KEGGs disease pathways. The numbers of pathways per each class category as defined by KEGG are shown in Table 2.3.

Evaluation Metrics

A goal class label was defined as a specific disease category that the binary classifier should report as positive example. For example, Cancer could be a goal class label, in contrast to the NonCancer class label, which should be reported as negative example. True positives (TP) were defined as instances with goal class label and to which the classifier assigned goal class labels. False negatives (FN) were defined as instances with goal class

label that were assigned non-goal class labels by the classifier. False positives (FP) were defined as instances with non-goal class labels to which the classifier assigned goal class labels. In this study, accuracy was measured as the geometric mean of Positive Predictive Value (PPV) and Sensitivity (S_n), where $PPV = \frac{TP}{TP+FP}$ and $S_n = \frac{TP}{TP+FN}$. Finally, accuracy was defined as: $A_g = \sqrt{PPVxS_n}$.

Table 2.3: KEGG disease pathway classes.

Disease Class	**Instances**
Cancer	15
Infectious	22
Substance Dependence	1
Neurodegenerative	5
Immune	7
Cardiovascular	4
Metabolic	3

Experiments

Each pathway in the dataset was annotated with GO terms of the manually curated HPRD database. By excluding the Substance Dependence pathway data, which had only one instance, this dataset was used to test six binary classifiers, one for each disease class. For each disease class, a two-label dataset was generated. For instance, a cancer dataset was developed that contained pathways with labels cancer and non-cancer. Then, evaluation of accuracy of each binary classifier was measured on these six datasets. A 3-fold cross validation procedure was applied to each dataset. Given the small dataset, cross validation procedure was run for 30 iterations and average accuracy was calculated.

2.4 Results

2.4.1 Classification Accuracy

Average accuracy for each of the datasets is shown in Table 2.5 (based on 30 cross validation runs). For Metabolic, Cardiovascular, Neurodegenerative, and Substance Dependence datasets, the system was not able to classify any positive classes correctly (TP value was zero), due to the small number of instances of these classes in dataset. However, the total number of instances of these classes was 20, therefore including them, as negative examples of cancer and infectious disease, in training data was important.

Table 2.4: Average classification accuracy.

Disease Class	**Accuracy**	**PPV**	S_n
Cancer	0.8	0.77	0.83
Infectious	0.67	0.6	0.75

2.4.2 Disease Fingerprints

As a by-product of the EM training process, the best partitioning of each pathway graph was saved to output files. Each partitioning highlighted a set of subgraphs (features) inside a pathway. Annotating nodes of pathways with functional annotations (e.g., GO terms) yielded an abstract representation of pathways. Thus, the subgraphs identified inside each pathway could be regarded as functional sub-units. Each category of disease was assumed to have its characteristic functional units (fingerprints) inside pathways under that category. Individual GO terms could be found equally in pathway graphs of two different disease

classes. Correlation tests may not be able to find a strong association between a disease class and a given GO annotation of genes in pathway graphs. However, the conditional probability distribution of paths and disease classes suggested that some paths tend to be found more frequently in a specific class of diseases and less frequently in other classes. Table 2.5 shows a number of paths that tend to appear in cancer pathways and those that tend to appear in non-cancer disease pathways.

Table 2.5: Paths associated with cancer/non-cancer.

Annotated Path	**Disease Class**
GO:0003924-activation-GO:0004674#	Cancer
GO:0004713-inhibition-GO:0004713#	Cancer
GO:0003924-activation-GO:0030159#-GO:0030159-activation-GO:0004674#	Cancer
GO:0003924-activation-GO:0004674#-GO:0004674-phosphorylation-GO:0004712#	Cancer
GO:0030528-dissociation-GO:0003700#	Non-Cancer
GO:0004713-phosphorylation-GO:0003700#	Non-Cancer
GO:0005509-binding/association-GO:0005200#-GO:0005200-binding/association-GO:0005198#	Non-Cancer
GO:0004930-activation-GO:0003924#-GO:0003924-indirect_effect-GO:0004620#	Non-Cancer

The maps shown in Figure 2.4-(e) give macro view of the linkage of GO terms in annotated graphs and demonstrate that distribution of pairs of GO terms is sparse. In general, this view does not suggest much about the structure of graphs. Instead, they reflect the fact that although there are many edges connecting GO terms in the graph dataset, only few GO term pairs are linked more frequently than other. However, structural patterns that

are uncovered in graphs can be important to learn facts about key functional components inside graphs. Figure 2.4 shows such structural patterns, which are linked to maps of basic processes of expression, phosphorylation, and activation as shown in Figure 2.4-(e). The mapping of edges of these structural patterns into maps in Figure 2.4 suggests that biological meaningful patterns do not necessarily correspond to spatial patterns in maps. This might be because functional similarity is not the only reason to link two genes in a given disease pathway. Functionally dissimilar genes might be found linked in a pathway, and thus it would be expected to find dissimilar (spatially distant) GO terms to be linked in a disease fingerprint (subgraph), but found spatially scattered in the map. It should also be mentioned that the method presented here allows for the inclusion of some edges in a disease fingerprint (subgraph) even though these edges are not directly related to that disease.

2.5 Discussion

Extracting meaningful structural patterns (fingerprints) of disease categories is important for many reasons. Meaningful patterns can illustrate interactions between proteins in functional terms that would help better understanding of genetic pathways. This, in turn, can help biomedical and pharmacological researchers indentify important biological sub-processes that might take place inside cells. From a knowledge discovery perspective, identifying sets of fingerprints of disease pathways can be important for mining tasks such as discovery and classification of disease pathways. In this study, a probabilistic model was developed to identify such important substructures (disease fingerprints) in functionally annotated pathways. Identified disease fingerprints were used in classification of test set of disease pathways into disease categories.

The synergy of different sources of biological knowledge bases and computational models is important to uncover patterns of interest. Biochemical, physicochemical, graph based properties are integrated in models of analysis of large biological networks. Using network properties alone can help in studying of structure and general dynamics of networks, while looking for useful and meaningful patterns would require incorporation of knowledge sources. Functional annotations have been shown important for discovery, analysis, and classification of genetic pathways based on biological functions (Cakmak and Ozsoyoglu 2007, Cerami et al. 2010, Hu et al. 2005, Liu et al. 2009). In this study, GO terms were used to enrich KEGGs disease pathway graphs with functional annotations, which were essential to represent these graphs at an abstract level. The integration of knowledge sources also requires specially designed computational models to make best

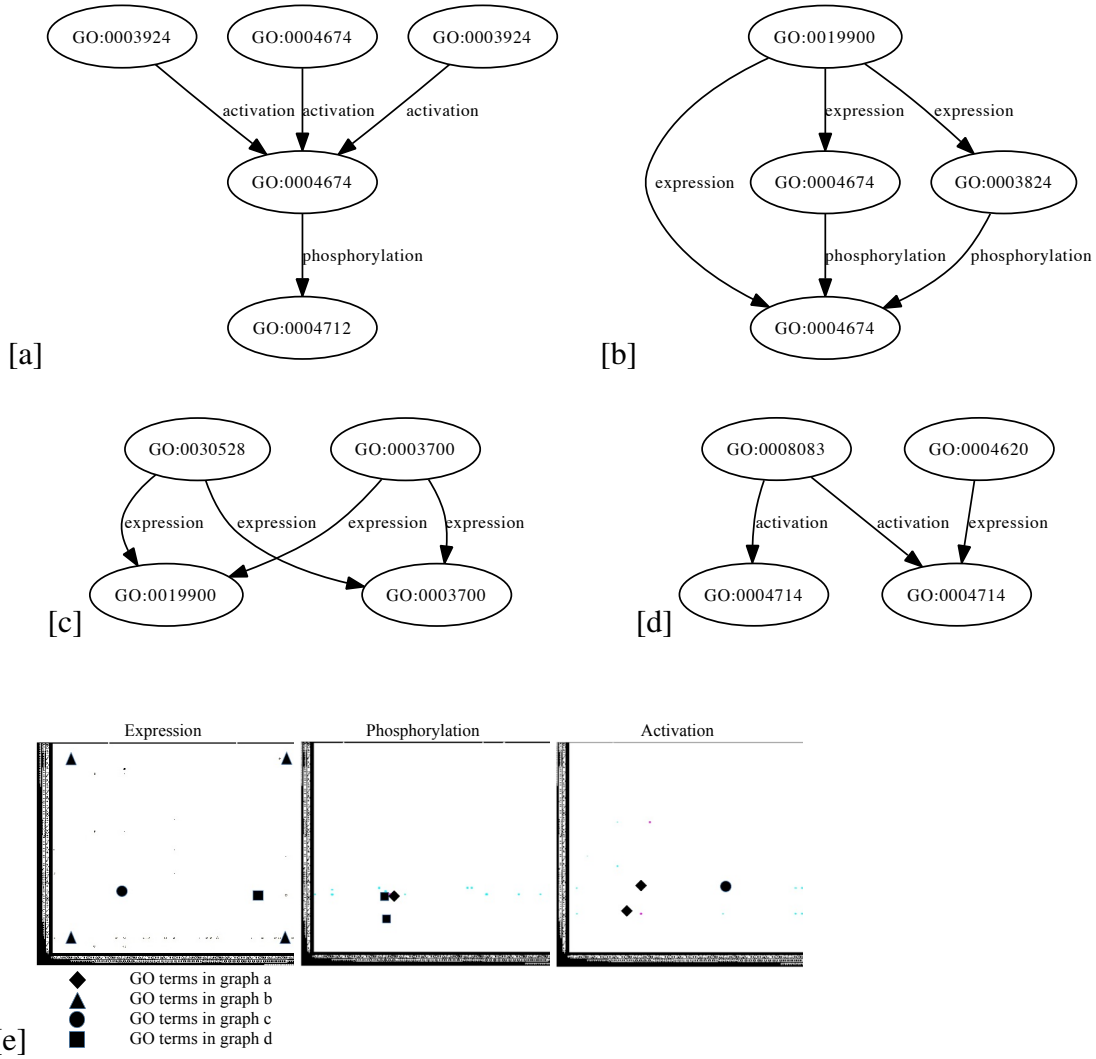


Figure 2.4: Disease fingerprints for cancer pathways and the mapping of their nodes onto maps that represent GO terms associations in data. Directed graphs that represent fingerprints extracted from best partitionings of cancer pathways are shown in (a)-(d). Pairs of GO terms in (a)-(d) that were part of expression, phosphorylation, and activation processes are highlighted in the maps shown in (e), with axes representing GO terms.

use of these sources. For instance, in this study, functional annotations were incorporated in a probabilistic model that took into account associations between sets of functional annotations as represented in paths and subgraphs. In contrast, using separate functional annotations as features could be less effective than expected. For instance, only small set of GO terms was identified as optimal features and was encoded in feature vectors for graph classifications of pathway diagrams (Huang et al. 2011). Knowledge-enriched models that make use of associations between GO terms can be more effective (Felsenstein 2004).

This study aimed to address a problem related to discovery of key structural patterns in graph datasets. These patterns were searched for in the training process of a graph classification model. The problem was cast as finding optimal substructure feature sets (fingerprints). The concept of partitioning enabled searching for features in a coherent way that is effective in avoiding irrelevant or redundant structural patterns. The proposed mathematical model and EM algorithm used the concept of partitioning to get better estimates for the conditional probability distribution for graph paths given disease classes. This idea can be similar to maximum likelihood (ML) phylogenetic analysis (Felsenstein 2004). In a sense, ML phylogenetic analysis uses nucleotide transition probability distribution to search for more likely phylogenetic trees (which can be perceived as a hierarchy). The ML training for phylogenetic analysis produces a best scoring phylogenetic tree for a set of genes while improving parameters values for nucleotide transition probability distribution. A similar practice was followed here: the study aimed to produce the best partitionings while improving the conditional distribution of paths given classes.

CHAPTER 2. MINING DISEASE FINGERPRINTS FROM WITHIN GENETIC PATHWAYS

Identifying optimal feature set for graph classification is an important problem in graph data mining (Jin et al. 2009, Ranu and Singh 2009). One method for graph classification is to use graph pattern mining to generate candidate features. Then, optimal feature set for classification is identified using variety of measurements such as information gain. However, graph classification techniques that use graph pattern mining for feature selection have three major problems:

1. *The search for features is local and sequential.* Candidate subgraph features are extracted and evaluated in isolation. The problem with this method is that features can be redundant or less informative.

2. *The criteria used for feature selection might not be optimal.* For instance, subgraph frequency can be used as criterion for selecting features (e.g., using gSpan (Yan and Han 2002) to find candidate features). Frequent subgraphs may not necessarily be discriminative. On the other hand, some information theoretic features may not be effective. For instance, LEAP search utilized information gain to look for features. This strategy may fail in the following scenario as noted by Jin, et al.(Jin et al. 2009): When no individual pattern has high discrimination power, a group of patterns may jointly have higher discrimination power. Since LEAP search finds patterns sequentially, it is unlikely that it will find such groups of jointly high discriminative power.

3. *It can be difficult to separate the searching and classification processes.*Separating the search for subgraph features and classification when using feature vectors can prevent the classification algorithm from using prior information about the distribution of class labels among graph instances.

To address the above problems in this study, the search for optimal feature set was integrated into the training of a probabilistic model for graph classification. The concept of partitioning and the utility of the partitioning function provided a means that naturally divided graphs into candidate features. Upon completion of model training, the best partitioning of each pathway instance provided a list of subgraphs that were considered characteristic components of a given pathway. The limited size of the design dataset and few number of instances per some disease classes made it not possible to analyze fingerprints for some disease classes. As more diseases have related processes identified in the future, it may be possible to analyze their fingerprints. Disease pathways in databases other than KEGG would be considered in a future work to overcome the limits of small dataset size. The scalability of this method to larger networks can be obtained by adjusting the maximum number of partitionings, which is an adjustable parameter of the tool as mentioned in the Methods section. By keeping smaller number of partitionings per graph instance, larger networks with increased annotations can be processed.

2.6 Conclusion

In this paper, an approach is presented for structural analysis and classification of genetic pathways of human diseases. Experiments on real data show good performance in terms of classification accuracy while identifying characteristic components inside each pathway both in training and testing examples. The highlighting of characteristic functional components (fingerprints) inside each pathway gives justification of classification decisions and may help improve the understanding of how genetic pathways act at component level. The proposed model may be generalized to many biological networks that are modeled as annotated directed graphs.

2.7 References

Arrell, D. and A. Terzic (2010). Network systems biology for drug discovery. *88*.

Barabasi, A., N. Gulbahce, and J. Loscalzo (2011). Network medicine: a network-based approach to human disease. *Nature Reviews Genetics 12*(1), 56–68.

Battle, A., M. Jonikas, P. Walter, J. Weissman, and D. Koller (2010). Automated identification of pathways from quantitative genetic interaction data. *Molecular Systems Biology 6*(1).

Brown, P. F., V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer (1993). The mathematics of statistical machine translation: Parameter estimation. *19*.

Cakmak, A. and G. Ozsoyoglu (2007). Mining biological networks for unknown pathways. *Bioinformatics 23*(20), 2775.

Caspi, R., H. Foerster, C. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S. Rhee, A. Shearer, and C. Tissier (2008). The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic acids research 36*(suppl 1), D623–D631.

Cerami, E., E. Demir, N. Schultz, B. Taylor, and C. Sander (2010). Automated network analysis identifies core pathways in glioblastoma. *PLoS One 5*(2), e8918.

Chen, L., T. Huang, X. Shi, Y. Cai, and K. Chou (2010). Analysis of protein pathway networks using hybrid properties. *Molecules 15*(11), 8177–8192.

Cogswell, J., J. Ward, I. Taylor, M. Waters, Y. Shi, B. Cannon, K. Kelnar, J. Kemppainen, D. Brown, and C. Chen (2008). Identification of mirna changes in alzheimer's disease brain and csf yields putative biomarkers and insights into disease pathways. *Journal of Alzheimer's disease 14*(1), 27–41.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–38.

Felsenstein, J. (2004). *Inferring phylogenies*. Sinauer Associates.

Franke, L., H. Bakel, L. Fokkens, E. De Jong, M. Egmont-Petersen, and C. Wijmenga (2006). Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *The American Journal of Human Genetics 78*(6), 1011–1025.

Goto, N., P. Prins, M. Nakao, R. Bonnal, J. Aerts, and T. Katayama (2010). Bioruby: Bioinformatics software for the ruby programming language. *Bioinformatics 26*(20), 2617.

Hu, H., X. Yan, Y. Huang, J. Han, and X. Zhou (2005). Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics 21*(suppl 1), i213.

Huang, T., L. Chen, Y.-D. Cai, and K.-C. Chou (2011). Classification and analysis of regulatory pathways using graph property, biochemical and physicochemical property, and functional property. *6*.

Jin, N., C. Young, and W. Wang (2009). Graph classification based on pattern co-occurrence. pp. 573–582. ACM.

Joshi-Tope, G., M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, and L. Matthews (2005). Reactome: a knowledgebase of biological pathways. *33*.

Kanehisa, M., S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa (2010). Kegg for representation and analysis of molecular networks involving diseases and drugs. *38*.

Karnovsky, A., T. Weymouth, T. Hull, G. Tarcea, G. Scardoni, C. Laudanna, M. Sartor, K. Stringer, H. V. Jagadish, C. Burant, B. Athey, and G. Omenn (2012). Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data. *Bioinformatics 28*(3), 373–380.

Karp, P., M. Riley, S. Paley, and A. Pellegrini-Toole (2002). The metacyc database. *Nucleic acids research 30*(1), 59–61.

Khatri, P., M. Sirota, and A. Butte (2012). Ten years of pathway analysis: Current approaches and outstanding challenges. *PLoS Computational Biology 8*(2), e1002375.

Lambert, J., B. Grenier-Boley, V. Chouraki, S. Heath, D. Zelenika, N. Fievet, D. Hannequin, F. Pasquier, O. Hanon, and A. Brice (2010). Implication of the immune system in alzheimer's disease: evidence from genome-wide pathway analysis. *Journal of Alzheimer's disease 20*(4), 1107–1118.

Liu, G., L. Wong, and H. Chua (2009). Complex discovery from weighted ppi networks. *Bioinformatics 25*(15), 1891.

Liu, N. and E. N. Olson (2010). Microrna regulatory networks in cardiovascular development. *18*.

Maudsley, S., W. Chadwick, L. Wang, Y. Zhou, B. Martin, and S. Park (2011). Bioinformatic approaches to metabolic pathways analysis. *Methods in molecular biology (Clifton, NJ) 756*, 99.

Novoyatleva, T., F. Diehl, M. J. Van Amerongen, C. Patra, F. Ferrazzi, R. Bellazzi, and F. B. Engel (2010). Tweak is a positive regulator of cardiomyocyte proliferation. *85*.

Pan, T., S. Kondo, W. Le, and J. Jankovic (2008). The role of autophagy-lysosome pathway in neurodegeneration associated with parkinson's disease. *Brain 131*(8), 1969.

Pawson, T. and R. Linding (2008). Network medicine. *582*.

Prasad, T., R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, and A. Venugopal (2009). Human protein reference database - 2009 update. *Nucleic acids research 37*(suppl 1), D767–D772.

Ranu, S. and A. K. Singh (2009). Graphsig: A scalable approach to mining significant subgraphs in large graph databases. IEEE.

Rual, J., K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. Berriz, F. Gibbons, M. Dreze, and N. Ayivi-Guedehoussou (2005). Towards a proteome-scale map of the human protein-protein interaction network. *Nature 437*(7062), 1173–1178.

Rudy, Y., M. J. Ackerman, D. M. Bers, C. E. Clancy, S. R. Houser, B. London, A. D. McCulloch, D. A. Przywara, R. L. Rasmusson, and R. J. Solaro (2008). Systems approach to understanding electromechanical activity in the human heart. *118*.

Slattery, M., R. Wolff, K. Curtin, F. Fitzpatrick, J. Herrick, J. Potter, B. Caan, and W. Samowitz (2009). Colon tumor mutations and epigenetic changes associated with genetic polymorphism: Insight into disease pathways. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis 660*(1-2), 12–21.

Stelzl, U., U. Worm, M. Lalowski, C. Haenig, F. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, and S. Koeppen (2005). A human protein-protein interaction network: a resource for annotating the proteome. *Cell 122*(6), 957–968.

Yan, X. and J. Han (2002). gspan: Graph-based substructure pattern mining. IEEE.

You, C., L. Holder, and D. Cook (2009). Substructure analysis of metabolic pathways by graph-based relational learning.

Chapter 3

GPAM: Graph Pattern Analysis Model

Nabhan, A. R. and I. N. Sarkar (2013). GPAM: Graph Pattern Analysis Model. In preparation.

3.1 Abstract

Structural pattern analysis of graph and network data is a core problem in graph data mining tasks including exploratory data analysis (e.g., detecting significant patterns), learning (e.g., clustering, classification), and data management (e.g., graph indexing and query). State-of-the-art graph pattern analysis methods (e.g., significant pattern mining and graph kernels) aim to map high dimensional graph data into low dimensional feature space. Key limitations of these methods include: (1) the search for patterns is local and sequential, and (2) selecting proper interestingness measure can affect performance. In this paper, a graph pattern analysis model (GPAM) is presented. This model allows for simultaneous and global pattern analysis of graphs in a dataset, taking into consideration the context of a candidate pattern (i.e., neighboring subgraph patterns). The iterative algorithm for learn-

ing model probabilities leads to emergence of significant patterns in the dataset. Efficacy of GPAM is demonstrated by implementing two graph classification systems for graph datasets of chemical compounds. Results show that the proposed model can be a viable alternative to current kernel-based and graph mining methods.

Index Terms: Structural Pattern Analysis, Graph classification, Graph Partitioning Function

3.2 Introduction

Learning from data of complex structures such as graphs has been a challenging task for the inference of new knowledge. This is mostly because modeling dependencies in complex data cannot trivially be performed at micro-level connections between nodes. For example, graph data are mainly made to describe structured and complex relations among a set of collective agents or objects. Basic attributes of graph elements (i.e., edge and node labels) cannot solely characterize the semantics inherently in graphs. Macro-level structural patterns (that are composed of combinations of basic graph elements) are hence more useful and informative in representing key information embedded in graphs. Meaningful dependencies should involve these macro-level patterns that graph capture semantics.

The use of structural pattern analysis methods can help gaining insight into structure of graph and network data. A range of methods related to graph data management (e.g., indexing and querying) and learning (e.g., clustering, classification) can utilize graph structural pattern analysis. Applications of structural pattern analysis of graphs are found in many domains, including: chemical informatics, bioinformatics, and graph

data stream analysis. In chemical informatics, analyzing graphs representing chemical compounds has important applications including prediction of biochemical characteristics using Quantitative Structural-Property Relationships (QSPRs) (Espinosa et al. 2000). Computational prediction of compounds properties is an important way to reduce search space of chemical compounds in drug design research (Brown et al. 2010). Target properties of chemical compounds can be defined using structural features. In bioinformatics, structural pattern analysis of graph and network data has many applications in prediction of protein functions using structural features in protein graphs, analysis and prediction of biochemical pathways, and analysis protein interaction networks. Graph kernels and graph pattern mining are two methods for analyzing graph data by mapping high dimensional graph data into a feature space that is more suitable for the learning task. Graph kernels are elegant mathematical models to map graphs into feature space by measuring the similarity between pairs of graphs in the dataset. Quantifying the similarity between graphs can utilize structural patterns such as subtree, cycles, and shortest paths. In graph mining, structural patterns are extracted within graphs and weighted by different interestingness measures such as information gain and entropy. Effective algorithms for pattern extractions have been developed, including gSpan (Yan and Han 2002) and LEAP search (Yan et al. 2008).

Despite the success of graph kernels and graph pattern mining techniques in analyzing graph data, there are a number of limitations related to both techniques. Graph kernels have two major challenges (Li et al. 2011):

1. Finding computationally tractable kernel functions.

2. For some graph kernels, it might be efficient to compute kernel functions, but at the cost of leaving out good structural patterns that express graph semantics (Borgwardt and Kriegel 2005).

On the other hand, extracting features using graph pattern mining has three major challenges:

1. The search for features is local and sequential. Candidate subgraph features are extracted and evaluated in isolation. This results in redundant or less informative features. Techniques for handling redundancy could be used to address this challenge.

2. The criteria used for feature selection might not be optimal. For instance, subgraph frequency can be used as criterion for selecting features (e.g., using gSpan algorithm, which is frequency-based graph miner). Frequent subgraphs may not necessarily be discriminative. On the other hand, some information theoretic features may not be discriminative for classification. For instance, LEAP search algorithm utilizes information gain to look for features. This strategy may fail in the following scenario, according to Jin, et al. (Jin et al. 2009): When no individual pattern has high discrimination power, a group of patterns may jointly have higher discrimination power. Since LEAP search finds subgraph patterns sequentially, it is unlikely that it will find such groups of jointly high discriminative power.

3. It can be suboptimal to separate the searching for features and the learning task.

In this paper, a new model for structural pattern analysis of graphs, the Graph Pattern Analysis Model (GPAM), is presented. The model is based on the concept of graph partitioning: a function that maps graphs into edge-disjoint subgraphs. A given partitioning of a graph instance highlights structural patterns related to a given class. Subgraphs

are approximated by the sets of maximal paths found inside. Then, the class-conditional distribution of graphs is expressed as class-conditional paths distributions. The proposed model can be used to analyze a dataset of uncategorized graphs with the objective to find a set of structural patterns or can be used in categorized datasets where the objective is to highlight significant subgraph patterns within graph items that lie under each category. The probability estimation procedure of the proposed model yields two outcomes: (1) the set of partitionings of each graph that highlights key structural patterns; and (2) a conditional probability model that quantifies the dependency between maximal paths and class labels. As a benchmark for the new method, two graph classification systems were developed: (1) A Bayes classifier, and (2) A support vector machine (SVM)-based classifier. The Bayes graph classifier, in addition to utilizing statistical dependencies between structural patterns and class labels, can incorporate prior information about class label distribution in the dataset and thus is able to handle both balanced and unbalanced dataset. The SVM-based classifier runs on feature vector representations for the graph dataset using subgraph patterns that are highlighted within graphs using GPAM model.

Contributions of this work are:

1. A proposed partitioning function that is defined on graphs to highlight key structural patterns and give the probabilistic model an access to embedded patterns in each graph;

2. A heuristic search function based on Estimation-Maximization (EM) algorithm is developed for model probability estimation. At each iteration, the algorithm aims to find better-scoring partitioning functions of each graph - according to the model probability estimates from the previous iteration, and then improves the model probability

estimates using information from the new set of best partitionings. This iterative process leads to emergence of global structural patterns across the dataset;

3. Elimination of the need for a required separate graph mining step a priori; and

4. A graph classifier that is built on this model and performance is compared to graph kernels and frequent pattern-based classifiers. The proposed graph classifier can handle balanced and imbalanced graph datasets.

This paper is organized as follows. In Section 3.3, a general overview of graph kernel and graph mining approaches is given. Section 3.4 includes details about the proposed mathematical model. In Section 3.5, experimental settings are stated and the results on seven datasets are reported. Discussion and related work are given in Section 3.6. Finally, conclusions and future work are presented in Section 3.7.

3.3 Related Work

3.3.1 The Graph Kernels Approach

Kernel based methods (e.g., support vector machines [SVMs] and kernel principal component analysis [KPCA]) have been successfully applied to a range of learning problems (e.g., classification and regression) including various data types (e.g., text, graphs, and genome sequences) (Muller et al. 2001). Graph kernels have been developed to address learning problems related to graph and network data, including structural pattern analysis and graph classification and clustering. Applications of graph kernels include pattern recognition (e.g., image classification (Harchaoui and Bach 2007)), chemical informatics (e.g., molecular fingerprinting) (Ralaivola et al. 2005) , and bioinformatics (e.g., analysis

biological networks (Borgwardt et al. 2007) and protein function prediction (Borgwardt et al. 2005)). Graph kernels can be used to transform complex structured (usually non-linearly separable) data into a feature space where transformed data can be separated approximately linearly (Ralaivola et al. 2005). With a rigorous mathematical formulation, core computations of graph kernels can be performed via operations of matrix algebra. A graph kernel $k(u, v)$ measures the similarity between two graph objects u and v. Similarity measures can computed based on common structural patterns that graph objects share considering topology and link information as well as node labels of pairs of graphs in the dataset. There have been previous research efforts to develop graph kernels that can accommodate various structural patterns that better represent semantics embedded in graphs. Examples of graph kernels include subgraphs, subtrees, cyclic patterns, pattern diffusion. Details on models, properties, and algorithms of graph kernels can be found in (Muller et al. 2001) and (Vishwanathan et al. 2010).

The structural pattern analysis model presented in this paper (described in the next section) can be compared to a class of graph kernels known as Marginalized Graph Kernels in which the similarity between two graphs is calculated taking into account the amount of labeled sequences of nodes or walks that the two graphs share (Tsuda et al. 2002). It can also be compared to graph kernels based on subgraphs or graphlets. A summary of some graph kernel methods are presented below.

Random Walk Kernel

The basic idea of random walk kernels (Gärtner et al. 2003) is to construct a direct product graph for two input graphs. Every node in the direct product graph then represents a pair

of nodes from the original input graphs. An edge in the direct product graph exists if and only if the corresponding nodes in the original graphs are connected. Random walking on a graph G is the process of generating sequences of vertices that are chosen according to a transition probability function on adjacency matrix. The probability function determines the next vertex to be picked given the identity of vertices chosen so far. A random walk on the direct product graph corresponds to a simultaneous random walk in the original graphs. The number of walks that they share quantifies the similarity between the original two graphs.

Subtree Kernels and Cyclic Pattern Kernels

Kernels based on subtrees and cyclic patterns have been defined to capture more semantic and structural information in a graph than with random walks. Subtree patterns in graphs are creating by setting a node as a root and then adding all nodes that can be reached in a certain number of steps called tree height (Ramon and Gärtner 2003). A subtree kernel based on Weisfeiler-Lehman test of isomorphism is a fast kernel that can scale to larger graphs (Shervashidze and Borgwardt 2009). Kernels based on cyclic patterns (Horv $et\ al.2004$) count the number of cycles shared by two graphs, limited to a predefined number of simple cycles, because computing general cycles is NP-hard.

Shortest-paths Kernels

Graph kernels based on shortest paths (Borgwardt and Kriegel 2005) have computational advantage since shortest paths can be found in polynomial time and at the same time can

express the inherit semantics in graphs. A first step toward shortest-paths kernel is to transform original graphs into shortest-paths graphs using Floyds algorithm (Borgwardt and Kriegel 2005, Horv $et\ al.2004$). Then the shortest-paths kernel can be defined on edges of the Floyd transformed graphs.

Graphlet Kernels

Graphlets are subgraphs with a small number of nodes. For a subgraph order k, count vectors of all possible connected subgraph of order k are used to measure the similarity of two input graphs (Shervashidze et al. 2009). Graphlet kernels have a scalability advantage to process large graphs while expressing similarity of graphs based on shared subgraphs. This can be similar to pattern mining approaches to graph classification (e.g. GraphSig (Ranu and Singh 2009)) where feature vectors of graphs represent significant subgraphs within items in the graph dataset.

3.3.2 The Graph Pattern Mining Approach

A second popular approach to dimensionality reduction of graph data is to extract a set of subgraphs (that meet certain criteria such as frequency threshold and statistical significance) that represent candidate features and then use traditional feature filtering techniques that aim at selecting individual features that their distribution correlates the distribution of class labels (Hall 1999, Yu and Liu 2004). A vector representation of subgraph features is then used to solve learning problems (e.g., clustering and classification) or data management problems (e.g., graph indexing.) Interestingness measures can be used to guide the search for informative features. Examples of feature interestingness

measures include mutual information, support, confidence, information gain, and Pearson correlation. Tan et al. have described 21 interestingness measures that can be used for searching for features (Tan et al. 2002).

There is a range of methods for managing and learning from graph data that perform subgraph mining and create feature vectors indexed by subgraphs extracted from the graph dataset. Various measures can be used to search for informative or significant subgraph features. For graph data management problems, vectors of subgraph features can be used to index graphs using R-trees (Shokoufandeh et al. 1999). gIndex is a graph indexing method that relies basically on frequent substructures in graphs (Yan et al. 2004). Trees and discriminative subgraph patterns were combined for effective graph indexing (Zhao et al. 2007).

For learning problems of graph data, subgraph pattern mining methods have been utilized. In graph clustering problems, informative subgraph patterns can be used as features in vector space where clustering techniques can be applied in this feature space (Seeland et al. 2010). In graph classification problems, classifiers that are built on subgraph patterns as features can outperform graph embedding and kernel based methods for graph classification in terms of accuracy and efficiency. The LEAP search algorithm for finding significant pattern features combined with SVM yields better results than kernel-based methods for graph classification (Yan et al. 2008). GraphSig is a scalable feature selection algorithm for graphs that mines significant subgraphs using local patterns inside subgraphs captured by random walks between nodes (Ranu and Singh 2009). GAIA is an evolutionary computation algorithm for mining significant subgraphs in graph

datasets. It should be mentioned that the importance of mining significant subgraphs goes beyond learning problems to data management tasks such as graph indexing (Yan et al. 2004). The gBoost classifier employs a model that uses frequent subgraph pattern mining and linear programming to solve the graph classification problem (Saigo et al. 2009).

3.4 Graph Pattern Analysis Model

The problem formulation of this paper is how to search for hidden structural patterns in graphs. The hypothesis being formulated is that there is statistical dependency between structural patterns in a graph and category of this graph. A new function called the partitioning function maps a graph to a set of edge-disjoint subgraphs. This function allows for the formulation of dependency between graphs and class labels through access to structural subgraph patterns.

3.4.1 Preliminaries and Notations

A labeled graph is defined as $G(V, E, L_V, L_E, \sum_V, \sum_E)$, with set of vertices V, set of vertex labels $\sum_V$, set of edges E and a set of edge labels $\sum_E$. A node labeling function $L_V : V \rightarrow \sum_V$ assigns labels from a node alphabet set $\sum_V$ to nodes and an edge labeling function $L_E : E \rightarrow \sum_E$ assigns labels from an edge alphabet set $\sum_E$ to edges. A labeled subgraph g consists of a subset of nodes of G and edges that link them. A given graph can be directed or undirected. Each graph instance in the design dataset is assigned a class

label from a set C of class labels. Subgraphs are defined by subsets of vertex set and edge set of a graph.

Definition 3.1 *(Partitioning π) Let $E(.)$ denote edge set of a graph G. A partitioning is a function $\pi : E(G) \rightarrow Z$ that assigns an integer to every edge of G such that edges with the same integer form a subgraph. The set of subgraphs H_π highlighted by a specific partitioning function π is defined as $H_\pi = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \pi(e) = i\}$.*

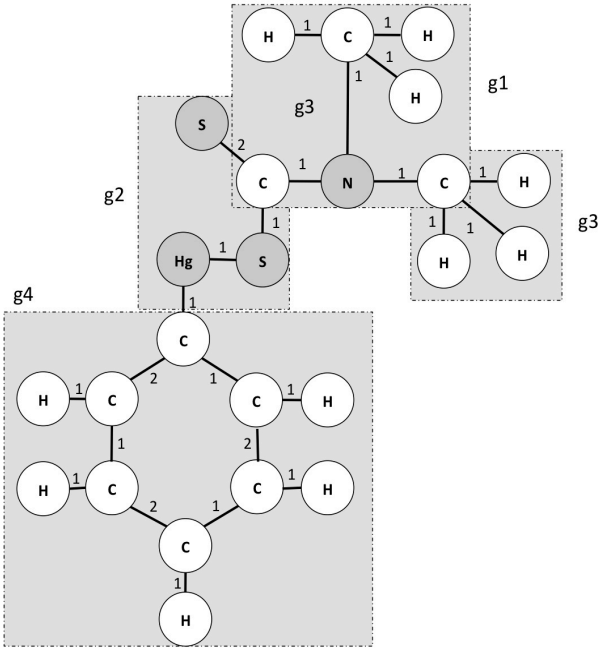


Figure 3.1: A chemical compound graph and a partitioning function that maps its edges into four subgraphs.

Figure 3.1 illustrates the concept of partitioning. According to the above definition, it follows that an edge belongs to exactly one subgraph in any given partitioning function (i.e., subgraphs resulting of a given partitioning function are edge-disjoint). There can be

a large set of possible partitioning functions[1], depending on the size of a graph's edge set. Searching for partitioning functions that highlight key features of a graph class was thus one of the objectives of this study. Partitionings were represented by integer arrays where indices represent edge identifiers and values points to subgraphs to which an edge belongs. Based on the notion of partitioning functions, a mathematical model was developed to systematically evaluate partitionings in order to identify highly probable partitionings that highlight key subgraph patterns.

3.4.2 Mathematical Model

Items in the graph dataset were assumed to be independent and identically distributed data. A probability value $P(G|C)$ was used to quantify the relation between a graph and its class label. Modeling the conditional probability model $P(G|C)$ directly is hard because: (1) graphs would have to be aligned with each other; (2) a dissimilarity metric would be required to count instances of each graph; and (3) a data sparseness problem will arise since there is a low probability of finding isomorphic instances of the same graph in a given dataset. Graphs are therefore typically broken up into smaller subgraphs that tend to occur frequently in the design set. The graph partitioning function is used to decompose a graph into a set of (hypothesized) subgraph features. The fact that there can be a large set of possible paritionings of a graph allowed for the systematic exploration of subgraph feature space isn a coherent way that takes into account feature context (this is in contrast with frequent pattern mining methods that extracts subgraph features separate of each

[1]The number of partitioning functions of a graph with an edge set of n elements is given by the Bell number B_n (Aigner 1999).

other).

Each class of graphs is assumed to have its characteristics that are represented by: (1) labels of nodes and edges; and (2) a set of partitionings that structurally highlights various topological patterns known to be associated with a given class. Since a partitioning is a function of graphs, the search for the best class to assign to unlabeled graph instance is equivalent to finding the best set of partitionings that best capture topological patterns in this graph that are features of this best class. Let $P(G, \pi|C)$ denote the probability of a partitioning π of graph G given a class label C. Since there are many possible ways to partition a single graph, the conditional probability $P(G|C)$ is calculated as the sum of all possible partitionings of a given graph G. Hence, partitionings are introduced as a hidden parameter into the conditional probability model:

$$P(G|C) = \sum_{\pi} P(G, \pi|C) \tag{3.1}$$

For notational convenience, let H_π be the set of subgraphs according to a partitioning function π of graph G: $H_\pi = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \pi(e) = i\}$

Assuming that subgraphs resulting from a partitioning function are conditionally independent, $P(G, \pi|C)$ can be written as:

$$P(G, \pi|C) = \prod_{g \in H_\pi} P(g|C) \tag{3.2}$$

The probability value $P(g|C)$ quantifies the likelihood that subgraph g is a characteristic or feature of class C. The conditional independence assumption made here can be mathematically plausible, noting that: (1) subgraphs according to a given partitioning

do not overlap (i.e., do not share common edges, according to Definition 1); and (2) this assumption applies to subgraphs generated according to one partitioning function (i.e., it is local to a specific partitioning, not for all combinations of subgraphs.) Using the set of partitionings, the space of possible features of a class in the graph dataset can be explored simultaneously in a coherent way. The likelihood of each subgraph in the dataset is calculated in a way that takes into account other subgraphs in partitionings of all graphs in the dataset. The probability that a subgraph g is a feature of class C can be obtained by tabulating the co-occurrences of subgraph g and class label C in the entire training set. However, each of the counts of co-occurrences of (g, C) pairs should be proportionally weighted by the probability of the partitioning to which they belong. It is assumed that the higher the frequency of a given (g, C) pair in the data, the more likely that g is a discriminant feature of class label C. The difference between this method of searching for features and existing methods (which look for features sequentially and locally) is that this method allows for searching for multiple features simultaneously and globally by investigating existing partitioning sets of each graph across the dataset. Subgraph features affect the likelihood of their neighboring features in the same partitioning and features belonging to other partitionings for that graph instance. Hence, search for characteristic patterns is performed taking pattern contexts into account.

Modeling $P(g|C)$ directly by counting co-occurrences of subgraphs and class labels can be, however, computationally expensive since subgraph enumeration is NP-hard (Kong et al. 2011). To make the model computationally feasible, each subgraph is approximated by a set of paths that link nodes inside that subgraphs. Using paths inside graphs has been successfully applied to graph mining problems. Nijssen et al. used frequent paths as a first

step in the search for frequent subgraph patterns (Nijssen and Kok 2004). Gudes et al. used sets of disjoint paths to address the problem of mining frequent subgraph patterns (Gudes et al. 2006). DePiero and Krout used length-r paths to approximate subgraph isomorphism (DePiero and Krout 2003).

In this study, a similar technique was used to approximate the probability of a subgraph using labeled maximal paths that can be identified within the subgraph boundaries. A maximal path is a path that is not a prefix of another path in a given subgraph. Each maximal path represented a sequence of labels of nodes and edges that lay in that path. Using paths to approximate subgraphs can have many advantages. In directed graph datasets (which usually represent processes with flow of information between nodes), using paths can capture essential sequences (chains) of steps. Here, these labeled paths are the basic building blocks in the model. This is in contrast to using subgraphs as basic building blocks (or smallest units) in classification tasks, which might hide internal (micro) interactions between nodes and preventing the learning algorithm to utilize this information. More importantly, data sparseness is minimized when using paths, which can yield better probability estimates. Let a denote a labeled path inside a subgraph. Then,

$$P(g|C) = \prod_{a \in g} P(a|C) \tag{3.3}$$

With the aid of the partitioning concept, the conditional probability $P(G|C)$ is reduced to a conditional distribution of maximal paths given class labels. The likelihood of a partitioning and a graph given a class label can thus be further expanded as

$$P(G, \pi|C) = \prod_{g \in H_\pi} \prod_{a \in g} P(a|C) \tag{3.4}$$

And the probability of a single partitioning π given a graph G and class label C is represented as:

$$P(G|\pi, C) = \frac{P(G, \pi|C)}{\sum_{\pi'} P(G, \pi'|C)} \tag{3.5}$$

Finally, $P(G|C)$ is expressed as

$$P(G|C) = \sum_{\pi} \prod_{g \in H_\pi} \prod_{a \in g} P(a|C) \tag{3.6}$$

Equations 3.1-3.6 cast the problem of searching for structural pattern features as a problem of estimating a conditional distribution of labeled maximal paths given graph classes, while maintaining a set of best partitionings for each graph instance highlighting features.

3.4.3 Iterative Procedure for Model Parameter Estimation

Estimation of the conditional probability $P(a|C)$ is performed with the aid of partitioning function. The objective of training is to search for partitionings of each graph in the data that maximize the likelihood probability of a graph given a class that is, $P(G|C)$, according to 3.1-3.6. By making use of the fact that the equation is a linear combination of $P(G, \pi|C)$ probabilities, only a subset of best partitionings (in terms of probability) can be considered. This does not significantly affect accuracy (since partitionings that are not considered tend to have lower probability value), but does have a substantial effect

on speed and memory. Better estimates of $P(a|C)$ can be obtained by directing search towards finding the set of best partitionings.

The conditional probability $P(a|C)$ could be estimated in a straight forward way if a set of partitionings of graphs are available a priori. In that case, for each partitioning, paths inside subgraphs could be extracted, and co-occurrences of these paths and the class label of the graph could be collected to estimate that conditional probability. The main question here is how to count the co-occurrence of paths and classes when the path belongs (simultaneously) to more than one subgraph according to different partitioning functions? One solution is to weigh each path-class count by the probability of partitioning to which that path belongs. This process is called collecting fraction counts, since each count of path-class pairs is discounted by partitioning probability. The idea of collecting fraction counts had been previously applied to machine learning problems such as statistical machine translation (Brown et al. 1993).

Counting Class-Path Co-occurrences

Finding a way to calculate path-class pair counts by utilizing the idea of fraction counts assumes that partitionings of graphs are known. Unfortunately, knowledge about possible ways of partitioning a graph is not always available a priori, and partitionings must be searched for and scored while estimating the conditional probability $P(a|C)$. Now, a circular argument is raised: to collect counts of path-class pairs, probabilities of partitionings $P(a|C)$ is needed to compute the probability of each partitioning of a graph, and partitioning probabilities are needed to collect counts and weigh them. To overcome this problem, an Estimation-Maximization (EM)-style algorithm was developed for this

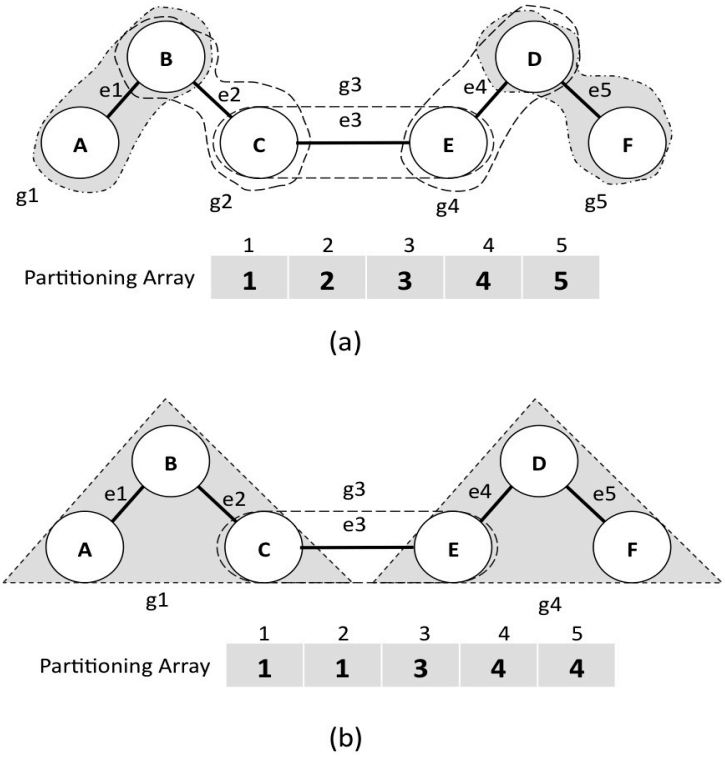


Figure 3.2: (a) A graph with six nodes and five edges with an initial partitioning mapping each edge to form one subgraph. (b) A new partitioning is formed in by merging edges e2 and e1 and edges e5 and e4. The resulting partitioning array contains values indicating three subgraphs: g1, g3, and g4.

study. The algorithm starts with an initialization step where seed partitioning vectors are randomly generated for each graph in the dataset. Counts of path-class pairs, (a, C), are then identified within each subgraph according to each partitioning. The counts of (a, C) pairs are normalized to create an initial conditional probability distribution $P(a|C)$. The next iteration of the algorithm starts with expanding existing partitionings of each graph to create more partitionings. Existing and new partitionings are scored using conditional probability $P(a|C)$ according to (3.6). Thus, the iterative training process has two steps: (1) E-Step: collecting fraction counts of (a, C) path-class pairs and computing (better)

estimates of a new conditional probability model $P(a|C)$; and (2) M-Step:searching for good partitionings for each graph and computing the probability of partitionings per graph using (3.4) and (3.5). For the dataset used in this study, three training iterations were run. The outline of this process is shown in Table 3.1.

At the start of training process, the conditional probability table $P(a|C)$ is initialized with short, single-edge paths. There is a minimum probability value (ϵ) for paths that are not discovered yet in early iterations of EM-style algorithm. In the M-Step, undiscovered paths are likely to be found when new partitionings (and probably larger subgraphs are likely to form and longer paths are found) are explored in the search for partitionings. These newly discovered paths are added to the conditional probability $P(a|C)$ when collecting fraction counts in the M-Step.

Searching for Highly Probable Partitionings

During model training, fraction counts of path-class pairs are collected from highly probable partitionings of graphs, leading to better estimates of the real conditional distribution $P(a|C)$. Therefore, the training procedure of the graph classifier should always look for and keep a set of best-scoring partitionings of each graph instance in the training datum throughout the iterations of the training algorithm. There are a large number of possible partitionings for a given graph instance, and it was therefore necessary for the system to limit the search by considering only a subset of best scoring partitionings. A parameter max_π was used to set the limit of the number of partitioning to keep in memory.

Each partitioning is represented as an array of integer numbers to map each edge (implicitly identified by its array index) to one subgraph. For illustration, let K be an array with length equal to the number of edges in a graph. Then, $K[i] = j$ means that edge number i in the edge set is mapped to subgraph j in the set of subgraphs of a given partitioning. To find a set of highly probable partitionings, the search process starts with a seed partitioning (with subgraphs containing only one edge) and then greedily expand a subgraph by adding edges from neighboring subgraphs (two subgraphs are neighbors if they share one or more vertices) to its set of edges. Starting with the integer array that represents a graph partitioning, two edges are randomly chosen and tested to see if they belong to different subgraphs. If this pair of edges is linked (share a vertex), then they are made to belong to the same subgraphs, meaning that the first edge is assigned the subgraph where the second edge resides. This step is repeated many times, resulting in the growing of some subgraphs and shrinking of others. Figure 3.2 shows this operation. The set of new partitionings is scored using (3.4) and (3.5). A priority queue was used to store partitionings ordered by their likelihood.

GPAM Application to Graph Classification

To test the efficacy of the mathematical model for structural pattern analysis, two graph classifiers were built: (1) a stand-alone Bayes classifier extending the mathematical model, and (2) a support vector machine (SVM) classifier that was run on features extracted from best partitionings. The graph classification problem is defined as, given a graph instance G, what is the best class that can be assigned to G from a model point of view. Mathematically, the problem is to maximize the probability $P(C|G)$. Using Bayes rule, $P(C|G)$ can be stated as

Table 3.1: An algorithm for model parameter estimation

Input:
D: graph data set $\{G_1, ..., G_n\}$
N: Number of iterations
Process
1: Create seed partitionings and Initialize $P(A|C)$ table with uniform probability value.
2: for i=1:N
E-Step
3: for each $G \in D$
4: for each $\pi \in G.\pi s$
5: for each $g \in S = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \pi(e) = i\}$
6: for each maximal path $A \in g$
7: $CountTable(A, C) += \pi.probability$
M-Step
8: Normalize entries of $CountTable(A, C)$ to obtain $P(A|C)$
9: for each $G \in D$
10: Let C be the class label of G.
Search for better graph partitionings: $G.\pi s$ = searchForPartitionings(G,C)
10: Use Eqs. 3.2-3.5 to compute the likelihood of every partitioning $\pi \in G.\pi s$
Output: updated $P(A|C)$, $G.\pi^*$ //return conditional probability and best partitioning

$$P(C|G) = \frac{P(C)P(G|C)}{P(G)} \tag{3.7}$$

Since choosing the best class C does not depend on P(G), (3.7) can be approximated by

$$P(C|G) \propto P(C)P(G|C) \tag{3.8}$$

Equation 3.8 casts the problem of graph classification into computing two probabilities: (1) $P(G|C)$ that represents important structural patterns in G that are characteristic of class

C; and (2) a prior knowledge about the distribution of class labels in the dataset. Having the prior probability $P(C)$ allows for handling unbalanced datasets.

Table 3.2: An algorithm for naïve Bayes' graph classification

Input: Graph G, set of class labels C, paths conditional probability distribution $P(A\|C)$ and prior class probability distribution $P(C)$
Process
1: For each class label $k \in C$
2: Using the conditional probability distribution $P(A\|C)$, π_s = searchForPartitionings(G,k)
3: Compute $P(k)P(G\|k)$ according to Eqs. 3.1-3.6 using the set of partitionings of G.
Output: Class label $k^*$with the maximum value of the product: $P(k^*)P(G\|k^*)$

Given a conditional probability model $P(a|C)$ for paths and class labels as well as a prior probability distribution model $P(C)$ for class labels, a new graph instance is assigned a class label as follows. A search for best partitioning for the target graph is started using positive and negative class labels. The evaluation of a partitioning quality is measured using $P(a|C)$ and $P(C)$ in (3.2-3.6). The class label that maximizes (3.8) is made output. The algorithm in Table 3.2 shows how classification is performed.

The SVM classifier operates on feature vectors consisting of subgraph patterns that were highlighted in best partitionings after running GPAM algorithm. Best partitionings were selected based on percentage of partitionings in the priority queue that the GPAM training algorithm uses for probability estimation. Two input parameters were used to control selection of subgraph features. The first parameter is the percentage of partitionings that are used to select subgraph features. The second input parameter sets the minimum

frequency of a subgraph pattern in order to include in the feature set.

3.5 Experiements And Results

3.5.1 Experimental Settings

GPAM was evaluated using two graph classification systems: (1) A Bayes graph classifier based on Equation 3.8 and (2) A GPAM+SVM classifier running on feature vector representations of graphs. The classification library libsvm (Chang and Lin 2011) was used as SVM classifier. The default Radial Basis Function (RBF) kernel in LIBSVM was used in all classification tasks in this study. Performance of the two systems was compared to a linear optimization graph classifier based on subgraph pattern mining, gBoost (Saigo et al. 2009), as well as graph kernel classifiers. Five graph kernel methods were used in the evaluation experiments: (1) Graphlet (G) kernel; (2) Ramon-Gartner subTree (RGT) kernel; (3) Weisfeiler-Lehman subTree (WLT) kernel; (4) fast geometric Random-walk (RW) kernel; and (5) Shortest Path (SP) kernel. The GPAM model was implemented using Java. MATLAB implementations (code and data for three of the datasets used in this study are made publically provided by Nino Shervashidze) were used for the graph kernels. The adjustable parameters of graph kernel methods were set as suggested in (Li et al. 2011). The classification library libsvm (Chang and Lin 2011) was used for learning from kernel matrices computed by each kernel method. Experiments were conducted at a parallelized computing facility using 56 processors (eight systems on seven datasets).

The software implementation of GPAM has two application parameters that needed to be set: (1) maximum subgraph size in each graph partitioning; and (2) the size of priority queue that stores partitionings of each item in the graph dataset. For this study, the maximum subgraph size was set to eight edges and the priority queue size was set to 100. For GPAM+SVM system, the proportion of partitionings that are used for feature extraction was set to 0.2, which means, for instance, that in a priority queue of 100 partitionings, the first 20 partitionings are used to extract subgraphs for feature vector representation of the dataset. For GPAM+SVM system, all graphs were treated as a single training dataset for pattern analysis and feature extraction and then the cross validation training and test sets were generated given the vector representation.

Following the evaluation methodology of paper by Li, et al. (Li et al. 2011), performance assessment was reported as average accuracy based on 10-fold cross validation run 10 times. While being faster and producing competitive accuracy compared to the five graph kernels mentioned above, the GF classifier developed by Li, et al. did not use structural features (e.g., trees or shortest paths in the graph). Therefore, the GF classifier was not assessed for performance in this study. In addition to reporting average accuracy, a two-sample t-test with equal variances at a 5% significance level was performed using 100 (10 fold run 10 times) data point of performance for GPAM and GPAM+SVM classifiers against the best performing classifier from graph kernels and gBoost methods. The null hypothesis (H_0) was that the mean accuracies of a pair of classifiers are equal. The alternative hypothesis (H_A) was that the means were not equal. The statistical analysis toolkit STATA was used to calculate accuracy results and performing analysis of variance

tests.

3.5.2 Dataset Description

A dataset of chemical compounds was used in the experimental evaluation of graph classification task. This dataset included Mutagenicity (MUTAG) of chemical compounds (Debnath et al. 1991), Predictive Toxicology Challenge (PTC) (Helma et al. 2001), National Cancer Institute (NCI) anti-cancer screening datasets: NCI1 and NCI109 (Wale and Karypis 2006). This benchmark data has been used previously in graph kernel evaluations (Li et al. 2011, Shervashidze et al. 2009). MUTAG is a dataset with class labels indicating mutagenicity of a chemical compound on bacterium Salmonella typhimurium. The NCI datasets are from National Cancer Institute and class labels of these two datasets indicate whether a compound is active or inactive based on an anticancer screen. Four datasets from the Predictive Toxicology Challenge (PTC) represent carcinogenicity of chemical compounds for Female Mice (FM), Male Mice (MM), Female Rats (FR) and Male Rats (MR). NCI1 and NCI109 data are balanced datasets (i.e., with roughly equal number of positive and negative instances). All other datasets are unbalanced. These datasets are undirected graphs with one label per node or edge.

3.5.3 Performance Evaluation

Table 2 shows the results of mean accuracy of the five graph kernels, gBoost and GPAM and GPAM+SVM methods. Each cell represents mean and standard deviation of accuracy for experiments of 10fold run 10 times. Results WL and RGT graph kernel methods for

Table 3.3: A description of chemical compounds datasets.

Dataset Identifier	**No. Items**	**No. Positive Classes**	**No. Negative Classes**	**Avg Nodes Per Item**	**Avg Edges Per Item**
MUTAG	188	125	63	18	39
NCI1	4110	2057	2053	30	32
NCI109	4127	2079	2048	30	32
PTC(FM)	349	143	206	25	25
PTC(MM)	336	129	207	25	25
PTC(FR)	351	121	230	26	26
PTC(MR)	344	152	192	25	26

NCI datasets are not available because the MATLAB program failed to finish within 24 hours. As shown in Table 2, there is a roughly steady decrease in the mean accuracy for all classifiers on data from the first dataset column (MUTAG) to the last dataset PTC_MR. This might suggest a decrease in feature richness going from MUTAG down to PTC datasets. The average accuracy of WL Tree kernel on the NCI109 dataset was the highest absolute average accuracy across all methods and datasets. The range of standard deviation values is larger for all datasets (approximately 78%), except for NCI1 and NCI109, with range of standard deviations 2.2.

On average, the MUTAG dataset has the highest standard deviation (and mean) value for the seven classification systems. Referring to Table 3.3, MUTAG has a skewed distribution of class labels (number of positive instances is roughly double that of negative instances). It is not clear whether this skewness of class label distribution or the inherent structural properties of MUTAG dataset that might be the cause of large standard deviation of accuracy for classifiers used in this study. The same statement could apply to the NCI1 and NCI109 datasets for which low standard deviation might be related to balanced class

distribution. To check the equal-variances assumption made for t-test statistical analysis, classifiers accuracy histograms for MUTAG and NCI1 datasets are shown in Figures 3.3-3.4 respectively. For instance, the variance in classification accuracy among the tested classifiers for the NCI dataset has roughly similar tight values as indicated by Figure 3.4.

The t-statistic values (Table 3.4) show that five out of seven accuracy results were found to be statistically significant. The WLT kernel achieved best results in three datasets; SP kernel achieved best result in one dataset; and the GPAM+SVM achieved best results in three datasets. It was noticed that Graphlet (G) kernel was the closest in performance to GPAM methods of all other kernel methods as shown in Table 3.4.

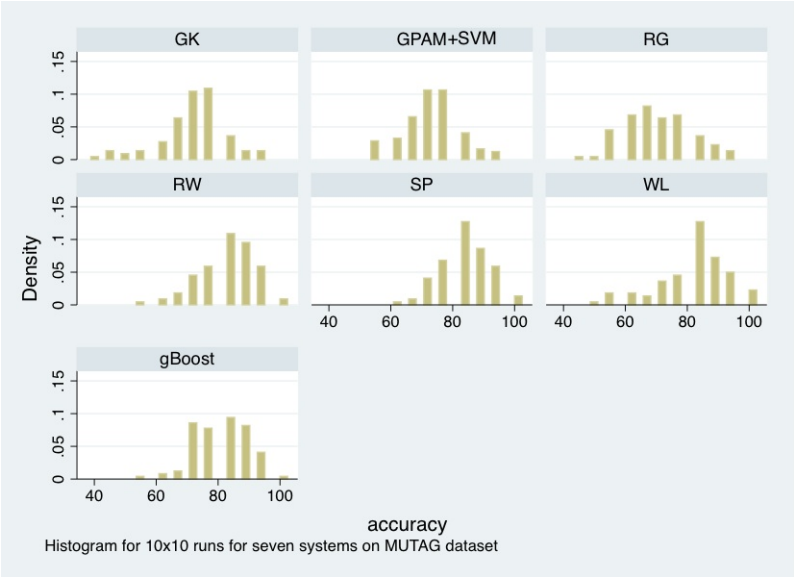


Figure 3.3: Classification accuracy histograms of MUTAG dataset.

3.6 Discussion

The proposed model for graph pattern analysis, GPAM, provides a way to search simultaneously for significant subgraphs by maintaining a set of best partitionings for each graph across that dataset. The core idea is to incorporate partitioning functions into a mathematical model that accounts for statistical dependency between a graph instance and its class label. The statistical dependency was reduced explicitly to a conditional probability model between maximal paths and class labels and implicitly by the set of partitioning functions that are scored by the conditional probability model.

One way to compare GPAM to graph kernels is to look at how essential graph semantics are captured during analysis. GPAM is similar to Shortest Path and Random

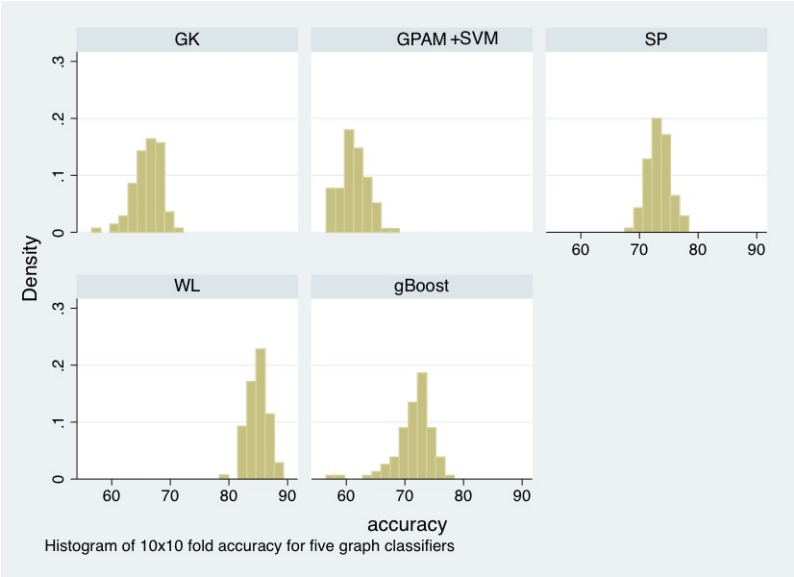


Figure 3.4: Classification accuracy histograms of NCI1 dataset.

Walk kernels in the way sequences of labeled nodes (maximal paths) are used to represents key structural patterns in graphs. One difference between GPAM and Random Walks and Shortest Path kernels is that GPAM sets boundaries on the labeled sequences by defining partitioning functions that map edges to subgraphs. A second difference is that similarity measurement between graphs is explicit (pair wise values) in graph kernels and implicit (graphs with similar partitionings have similar likelihood scores) in GPAM. For graph kernels, analysis is pair wise, and only information embedded in pairs of graphs for which the kernel is computed are relevant. Moreover, significant structural patterns in GPAM are highlighted within each graph instance across the entire dataset, while no patterns are identified in graph kernels. The search for key patterns using GPAM is performed collectively on all graphs under the same class.

Compared to graph mining-based approaches to map graphs into feature space, the proposed method avoids the decoupling of feature generation (using pattern mining) and feature filtering (using interestingness measures) by iterative improvement of probability model of maximal paths while maintaining best set of features that are most explicable by the model. Thus, GPAM provides a coherent framework for pattern searching. No post-processing or feature filtering is required. This search strategy avoids local, sequential pattern mining by taking into account neighbors of each subgraph features in the same partitionings. Significant subgraph features emerge during repetitive steps of the EM-like algorithm for probability estimation. The probability estimation algorithm considers large set of partitionings of graphs when computing probability values. One advantage of maintaining multiple partitionings per each graph is to allow for flexibility to consider feature overlap. Approximating subgraph matching using maximal paths allows for model flexibility to account for new graph instances in test dataset.

The GPAM method was benchmarked through the development of a Bayes graph classifier and SVM-based classifier using feature vectors built from significant subgraphs highlighted in best partitionings produced by training algorithm. Performance of these two GPAM-based classifiers was compared to graph kernels and graph mining-based classifiers. While Bayes classifier did not provide the overall best classification accuracy for any dataset, GPAM boosted with SVM classifier achieved best results for three datasets. Performance evaluation of the GPAM classifier shows that it can outperform graph kernels and graph mining-based classifiers for some datasets. The GPAM+SVM system achieved performance better than at least two graph kernels within each dataset. gBoost achieved better performance than GPAM+SVM on NCI1 and NCI109 datasets, while GPAM+SVM

outperformed gBoost on five datasets. The t-test statistic for two datasets (PTC_FM and PTC_MR) did not show a significant difference in performance between WLT kernel classifier and GPAM+SVM. It is also clear that, except for NCI1 and NCI109 datasets, there was large variability in accuracy results in each fold, indicating some inhomogeneity in distribution of significant features in the datasets.

There are a number of limitations of the proposed method. One limitation is in the definition of the partitioning function to divide graphs into edge-disjoint subgraphs. On the other hand, this edge-disjoining restriction allow for simple vector representation of partitionings as integer array. This allowed for easy modification of an existing partitioning by simple operations to change edge number (and hence, change the edge-to-subgraph membership.) A second limitation is the approximation of subgraphs by using maximal paths. However, this approximation has a computational advantage: there are polynomial-time algorithms for extracting paths inside subgraph and comparing paths is much easier than comparing subgraphs. The downside of this approximation is the inevitable partial decrease of accuracy. A third limitation is that the analysis is performed local to the class for which graphs are assigned. This makes it hard to discriminate very similar graphs of different class labels since the proposed method is designed to find similarity between graphs of the same class, not to maximize the distance between graphs of two different classes.

The inclusion of a partitioning function, as a variable in the computation of statistical dependency between a graph instance and its associated class label, is a new application of a successful idea that was previously applied in domains ranging from Natural Language

Processing (NLP) to evolutionary biology. In NLP, word alignment is a variable defined on a pair of a sentence and its translation to another language. The concept of word alignment allows for access to hidden structures in sentence pairs and is the main structure used by methods for building bilingual word dictionaries and word reordering models in statistical machine translation (Brown et al. 1993). The analogy here is that graph partitioning plays the same role as word alignment and the conditional probability model for maximal paths given classes is equivalent to the conditional probability of the word pairs in a bilingual dictionary. The EM algorithm is applied in both applications. The same analogy can be found in evolutionary biology when searching for the best phylogenetic tree for a set of gene sequences. In that context, graph partitioning is similar to the phylogenetic tree and the maximal paths probability model is analogous to the DNA base transition probability model that accounts for evolutionary events like mutations (Felsenstein 2004).

The results of the GPAM method call for three improvements of the model as part of future work. First, an investigation of other potential functions that allow access for hidden graph features is one way to extend the current study. Second, an alternative idea for approximating subgraphs by maximal paths is to use random walk inside subgraphs. The set of random walks can capture useful semantic features of graph by utilizing node and edge labels. Third, it may be useful to investigate searching techniques for exploration of space of partitioning functions. For instance, evolutionary computation may be useful methods to search for partitionings. The fitness function in this case would be the partitioning probability function as defined in Eq. 3.2. The integer array representation of partitionings can be used to represent individuals of populations and crossover and

mutation operations can be defined on this representation.

3.7 Conclusions

GPAM is a new model for structural pattern analysis of graphs. The search for significant subgraphs is performed globally across all graphs in the dataset under specific category and simultaneously by taking into considering neighboring subgraphs within the same partitioning, resulting in significant subgraphs emerge during model training. The model is also flexible with respect to the analysis of new data items, as the main data entity of the learned model is a conditional probability distribution of maximal paths. GPAM shows comparable performance when compared to previously described graph kernels and graph mining-based classifiers. For some datasets, the GPAM classifier outperforms current graph kernel methods and graph mining-based methods. The GPAM classifier reports the best partitionings for graphs in test data, therefore better justifying classification decisions.

3.8 References

Aigner, M. (1999). A characterization of the bell numbers. *Discrete mathematics 205*(1), 207–210.

Borgwardt, K. and H. Kriegel (2005). Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on*, pp. 8 pp. IEEE.

Borgwardt, K., H. Kriegel, S. Vishwanathan, and N. Schraudolph (2007). Graph kernels for disease outcome prediction from protein-protein interaction networks. In *Proc. of Pacific Symposium on Biocomputing (PSB)*, Volume 12, pp. 4–15.

Borgwardt, K., C. Ong, S. Schnauer, S. Vishwanathan, A. Smola, and H. Kriegel (2005). Protein function prediction via graph kernels. *Bioinformatics 21*(suppl 1), i47–i56.

Brown, J., T. URATA, T. TAMURA, A. MIDORI, T. KAWABATA, and T. AKUTSU (2010). Compound analysis via graph kernels incorporating chirality. *Journal of Bioinformatics and Computational Biology 8*(supp01), 63–81.

Brown, P. F., V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer (1993). The mathematics of statistical machine translation: Parameter estimation. *19*.

Chang, C. and C. Lin (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST) 2*(3), 27.

Debnath, A., R. Lopez de Compadre, G. Debnath, A. Shusterman, and C. Hansch (1991). Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry 34*(2), 786–797.

DePiero, F. and D. Krout (2003). An algorithm using length-r paths to approximate subgraph isomorphism. *Pattern recognition letters 24*(1), 33–46.

Espinosa, G., D. Yaffe, Y. Cohen, A. Arenas, and F. Giralt (2000). Neural network based quantitative structural property relations (qsprs) for predicting boiling points of aliphatic hydrocarbons. *Journal of Chemical Information and Computer Sciences 40*(3), 859–879.

Felsenstein, J. (2004). *Inferring phylogenies*. Sinauer Associates.

Gärtner, T., P. Flach, and S. Wrobel (2003). On graph kernels: Hardness results and efficient alternatives. *Learning Theory and Kernel Machines*, 129–143.

Gudes, E., S. E. Shimony, and N. Vanetik (2006). Discovering frequent graph patterns using disjoint paths. *18*.

Hall, M. (1999). *Correlation-based feature selection for machine learning*. Ph. D. thesis.

Harchaoui, Z. and F. Bach (2007). Image classification with segmentation graph kernels. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE.

Helma, C., R. King, S. Kramer, and A. Srinivasan (2001). The predictive toxicology challenge 2000äì2001. *Bioinformatics 17*(1), 107–108.

Horv °th, T., T. G§rtner, and S. Wrobel (2004). Cyclic pattern kernels for predictive graph mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 158–167. ACM.

Jin, N., C. Young, and W. Wang (2009). Graph classification based on pattern co-occurrence. pp. 573–582. ACM.

Kong, X., W. Fan, and P. S. Yu (2011). Dual active feature and sample selection for graph classification. ACM.

Li, G., M. Semerci, B. Yener, and M. Zaki (2011). Graph classification via topological and label attributes. In *9th Workshop on Mining and Learning with Graphs (with SIGKDD)*.

Muller, K., S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks 12*(2), 181–201.

Nijssen, S. and J. Kok (2004). A quickstart in frequent structure mining can make a difference. pp. 647–652. ACM.

Ralaivola, L., S. Swamidass, H. Saigo, and P. Baldi (2005). Graph kernels for chemical informatics. *Neural Networks 18*(8), 1093–1110.

Ramon, J. and T. Gärtner (2003). Expressivity versus efficiency of graph kernels. In *First International Workshop on Mining Graphs, Trees and Sequences*, pp. 65–74.

Ranu, S. and A. K. Singh (2009). Graphsig: A scalable approach to mining significant subgraphs in large graph databases. IEEE.

Saigo, H., S. Nowozin, T. Kadowaki, T. Kudo, and K. Tsuda (2009). gboost: a mathematical programming approach to graph classification and regression. *Machine learning 75*(1), 69–89.

Seeland, M., T. Girschick, F. Buchwald, and S. Kramer (2010). Online structural graph clustering using frequent subgraph mining. *Machine Learning and Knowledge Discovery in Databases*, 213–228.

Shervashidze, N. and K. M. Borgwardt (2009). Fast subtree kernels on graphs. In *Advances in Neural Information Processing Systems*, pp. 1660–1668.

Shervashidze, N., S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt (2009). Efficient graphlet kernels for large graph comparison. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics. Society for Artificial Intelligence and Statistics*.

Shokoufandeh, A., S. Dickinson, K. Siddiqi, and S. Zucker (1999). Indexing using a spectral encoding of topological structure. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, Volume 2. IEEE.

Tan, P. N., V. Kumar, and J. Srivastava (2002). Selecting the right interestingness measure for association patterns. ACM.

Tsuda, K., T. Kin, and K. Asai (2002). Marginalized kernels for biological sequences. *Bioinformatics 18*(suppl 1), S268–S275.

Vishwanathan, S., N. N. Schraudolph, R. Kondor, and K. M. Borgwardt (2010). Graph kernels. *The Journal of Machine Learning Research 99*, 1201–1242.

Wale, N. and G. Karypis (2006). Comparison of descriptor spaces for chemical compound retrieval and classification. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pp. 678–689. IEEE.

Yan, X., H. Cheng, J. Han, and P. Yu (2008). Mining significant graph patterns by leap search. pp. 433–444. ACM.

Yan, X. and J. Han (2002). gspan: Graph-based substructure pattern mining. IEEE.

Yan, X., P. Yu, and J. Han (2004). Graph indexing: a frequent structure-based approach. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 335–346. ACM.

Yu, L. and H. Liu (2004). Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research 5*, 1205–1224.

Zhao, P., J. Yu, and P. Yu (2007). Graph indexing: tree+ delta¡= graph. In *Proceedings of the 33rd international conference on Very large data bases*, pp. 938–949. VLDB Endowment.

Table 3.4: Mean accuracy ± standard deviation for each classifier on seven datasets (t-statistic values in bold to indicate statistically significant results compared second best classifier for the same dataset).

	MUTAG	NCI1	NCI109	PTC_FM1	PTC_FM	PTC_FR	PTC_MR
GPAM	73.16±8.97	61.2±2.28	62.75±8.22	58.32±8.02	62.75±8.22	64.80±6.88	56.59±5.94
GPAMSVM	81.22±8.78	69.22±3.11	67.13±2.69	**63.64±8.60**	**68.78±6.21**	**67.74±3.90**	60.08±8.22
G	72.03±10.84	66.12±2.35	66.69±2.04	59.21±8.46	64.44±7.58	66.09±7.04	57.12±8.75
RGT	70.43±10.85	–	–	62.28±8.04	64.47±8.15	67.01±7.71	58.16±8.13
WLT	81.85±11.29	**84.94±1.71**	**85.21±1.43**	62.81±8.20	66.43±7.65	67.55±7.22	**61.43±8.30**
RW	83.02±9.06	–	–	61.86±7.94	63.56±7.90	67.26±8.10	57.02±7.41
SP	**84.01±7.93**	73.24±1.85	73.05±2.25	59.90±8.75	62.72±8.82	64.47±7.55	59.70±7.53
gBoost	81.11±8.65	71.43±3.12	71.18±3.02	57.76±7.73	60.03±8.70	60.91±7.64	59.14±7.77
ttest	**2.2891**	**42.51**	**56.92**	0.6850	**2.3342**	**0.2287**	1.1251

Chapter 4

Structural Network Analysis of Biological Networks for Assessment of Potential Disease Model Organisms

Nabhan, A. R. and I. N. Sarkar (2013). Structural network analysis of biological networks for assessment of potential disease model organisms. *Journal of Biomedical Informatics. In press.*

4.1 Abstract

Model organisms provide opportunities to design research experiments focused on disease-related processes (e.g., using genetically engineered populations that produce phenotypes of interest). For some diseases, there may be non-obvious model organisms that can help in the study of underlying disease factors. In this study, an approach is presented that leverages knowledge about human diseases and associated biological interactions networks

to identify potential model organisms for a given disease category. The approach starts with the identification of functional and interaction patterns of diseases within genetic pathways. Next, the characteristic patterns are matched to interaction networks of candidate model organisms to identify similar subsystems that have the disease characteristic patterns. The quality of a candidate model organism is then determined by the degree to which the identified subsystems match genetic pathways from validated knowledge. The results of this study suggest that non-obvious model organisms may be identified through the proposed approach.

4.2 Introduction

Complex diseases stem from an interplay of genetic and environmental factors. At the genetic level, these diseases are often associated with the dysfunction of more than one gene. This necessitates the study of complex diseases at a systems level, which includes the modeling of cellular processes that underlie an observed disorder and may involve both sequential and simultaneous molecular interactions between many agents (e.g., genes and chemical compounds). This highlights the importance of curating molecular interaction networks (e.g., gene/protein interaction networks, metabolic networks, and genetic pathways). Data resources that catalogue these networks are increasing both in terms of the number and size of networks as well as their coverage of organisms. Environmental factors, on the other hand, complicate the study of human diseases, since it is difficult to create a controlled environment that enables scientists to study environmental effects on disease development. Hence, model organisms offer opportunities for detailed study of features associated with complex diseases, because these organisms may be genetically

engineered to produce desired phenotypes (e.g., associated with a particular disease of interest) and can be studied more easily in a controlled environment.

Model organisms play a vital role in advancing knowledge about disease processes. The sophisticated genetics of human diseases makes it important to study model organisms to uncover underlying mechanisms of diseases. Model organisms may not necessarily be closely related to humans from an evolutionary perspective. For instance, yeast are regularly used to model disease states (Aitman et al. 2011). Comparison of different phenotypes that arise from a conserved set of genes can be important for exploring model organisms for specific human disorders or diseases (Thomas et al. 2011, McGary et al. 2010). Analysis of model organism microarray data may also help identify those that have disease-related genes differentially expressed (Thomas et al. 2011).

The house mouse (Mus musculus) has been a typical model organism in the study of human disease processes (Bedell et al. 1997), as well as complex traits and social behavior (Koteja et al. 1999). Mice have also been genetically engineered to provide models for studying cancer and immune diseases (Haldar et al. 2007, Haldar et al. 2008). However, mice may not always be suitable for the study of all categories of disease. In a recent study of phenologs (phenotypes that are equivalent across organisms), McGary, et al. suggested a worm model (Caenorhabditis elegans) for breast cancer, a mouse model for autism, a plant model (Arabidopsis thaliana) for Waardenburg syndrome, and a yeast model (Saccharomyces cerevisiae) for angiogenesis disorders (McGary et al. 2010). Thus, there may be many potential choices for a suitable model organism relative to the spectrum of phenomena associated with disease. An empirical approach may therefore facilitate the

identification of organism(s) that might provide insights to human diseases.

Evaluation of candidate model organisms might be measured by the degree to which gene/protein interaction networks include pathways that are structurally and functionally similar to human disease-related biological processes. To this end, prediction of pathways in candidate model organisms that are similar to disease-related pathways in humans can be effective in evaluating model organisms. Pathway prediction can be performed by a variety of techniques. A widely used technique involves mining gene or protein interaction networks to extract dense subgraphs (highly connected components within the network) and then calculating the statistical significance of the discovered subgraphs (Ferrer et al. 2011). Statistically significant subgraphs are then cast as predicted pathways. Tian, et al. developed a method to discover statistically significant pathways from gene expression data (Tian et al. 2005). Bebek and Yang annotated gene networks with GO annotations and developed the PathFinder method to predict novel pathways (Bebek and Yang 2007). Cakmak and Ozsoyoglu developed a method that used frequent functional patterns in a known pathway to find organism-specific versions of that pathway in the gene networks (Cakmak and Ozsoyoglu 2007). Finally, Senf and Chen developed a hidden Markov model-based method to identify genes participating in genetic pathways (Senf and Chen 2009).

The present study proposes a computational method that attempts to provide a quantitative measure of how well a candidate model organism might be suited for the study of a given disease type. The proposed quantitative measure is based on the proportion of correctly predicted genetic pathways that can be identified in interaction networks for a

given organism. The proposed approach makes use of three types of knowledge resources: (1) Kyoto Encyclopedia of Gene and Genomes (KEGG) (Kanehisa et al. 2010) pathway database, (2) The Biological General Repository for Interaction Datasets (BioGRID) and Search Tool for the Retrieval of Interacting Genes/Proteins (STRING) gene/protein interaction databases (Stark et al. 2006), and (3) Gene Ontology (GO) (Ashburner et al. 2000) annotations that have been applied to genes or proteins in curated databases. The main premise of this work was to leverage a machine learning method to extract significant functional and structural patterns, or fingerprints, (Nabhan and Sarkar 2012) from functionally annotated KEGG disease pathways and match these patterns to functionally annotated gene/protein interaction networks in major databases (e.g., BioGRID) as well as meta-databases (e.g., STRING). Depending on an organisms interaction network coverage of structural patterns for a given disease, it can be ranked in terms of model organism suitability for that disease. Through the use of a statistical model, this study was able to quantify the dependency of functional structural patterns in pathways and disease categories for 14 organisms. It was assumed that some species may be a better suitable model for one disease category and thus less suitable for studying other diseases. This assumption was motivated by the McGary, et al. study, where a range of model species were suggested for complex diseases (McGary et al. 2010). The promising results suggest that the described approach may be used to determine the potential for a given organism to serve as a model for the study of a particular disease.

4.3 Materials and Methods

In this section, the five phases of the developed approach are described: (1) annotation of gene/protein nodes in pathway graphs with molecular function annotations, (2) learning

disease fingerprints within annotated pathways, (3) functional annotation and indexing of gene/protein interaction networks, (4) prediction of novel subsystems within gene/protein interaction networks using learned fingerprints, and (5) scoring discovered subsystems using reference pathways. Figure 4.1 provides an overview of the approach.

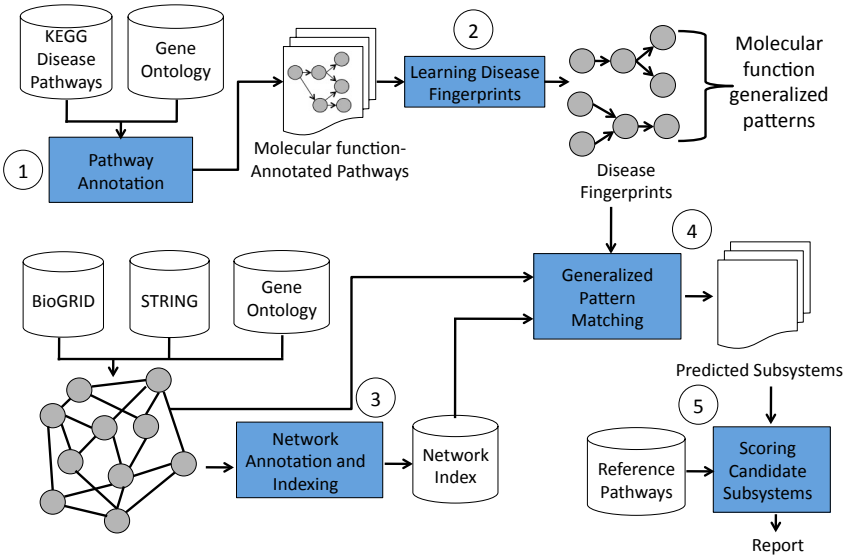


Figure 4.1: Overview of the five components of the method developed in this study.

4.3.1 Functional Annotation of KEGG Pathways

KEGG genetic pathways are modeled as directed graphs with a node set (V) representing biochemical entities such as genes, chemical compounds, and protein complexes and an edge set (E) representing interaction relations between entities such as general process type (e.g., a gene expression [GErel] or protein interaction [PPrel] relation) and specific relation types (e.g., activation, expression, and inhibition). For this study, only gene/protein nodes were considered. To increase the generalization capability, gene nodes were enriched with

molecular function annotations as defined in Gene Ontology (GO) (Ashburner et al. 2000). These GO annotations were imported from Human Protein Reference Database (HPRD) (Prasad et al. 2009) and overlaid on gene/protein nodes of pathway graphs. Gene/protein nodes without a match to HPRD GO annotations were assigned a default NULL annotation. Nodes could be associated with multiple GO term annotations and edges could also have multiple labels. Thus, for each graph there was a shift of focus from what gene/protein is in a given node? to what function does the node perform in a system that models a biological process? With knowledge-enriched annotations of genes/proteins, pathways were represented at a functional level. Subsequently, functional structural patterns in these pathways graphs could be matched to sub-networks of large interaction networks with functionally annotated nodes. In this study, the KEGG disease pathways dataset contained 63 disease pathways across seven human disease classes. KEGG disease pathways cover many biological processes related to genetic information processing, metabolism, and cellular processes. However, this study did not focus on a particular pathway category such as metabolic pathways and cellular processes. Each graph instance in this design set was associated with a class label from the seven disease classes in KEGG.

4.3.2 Learning Disease Fingerprints

The objective of the second module of the proposed method was to identify characteristic biological functionality patterns, termed fingerprints, in annotated disease pathways. A mathematical model and an algorithm were designed to accomplish this task. A disease fingerprint was defined as a subgraph within a GO annotated disease pathway. Fingerprints were assumed to represent functional sub-processes that could be characteristic of a disease class such as immune, infectious, or neurodegenerative disease. Graphs in the

design dataset were assumed to be independent and identically distributed (iid) data observed from an unknown probability distribution P(G). The iid data assumption was made to facilitate statistical inference and to make decision about properties (e.g., class label) of a graph instance independent of other graph instances in the dataset. For a given GO-annotated pathway graph, there can be a large number of possible GO functionality subgraph patterns, which will be called subgraph patterns hereafter. A mathematical model was proposed to allow for scoring of subgraph patterns. High scoring patterns were output from the model as disease fingerprints.

Mining of key subgraph patterns in the dataset was performed so that a subgraph pattern is evaluated within a context of its neighboring patterns in a graph. To formalize the idea of neighbor context, a utility function termed graph partitioning function was used to decompose a graph into a set of subgraphs. A partitioning function $\pi : E(G) \rightarrow Z$ assigned an integer to every edge e of graph edge set $E(G)$ such that edges with the same integer formed a subgraph. The set of subgraphs H that were highlighted by a specific partitioning function (π) was defined as $H_\pi = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \pi(e) = i\}$. Figure 4.2 illustrates the concept of partitioning.

Typically, there exists a large space of possible partitionings for a given graph. Searching for the most likely partitionings in the dataset leads to the identification of key subgraph patterns (fingerprints). Searching for best graph partitionings can be better than searching for individual subgraph patterns (e.g. as in frequent pattern mining techniques (Yan and Han 2002)). This is because a graph partitioning hypothetically decomposes a system (represented by a graph) into a set of components (subgraphs), and partitioning

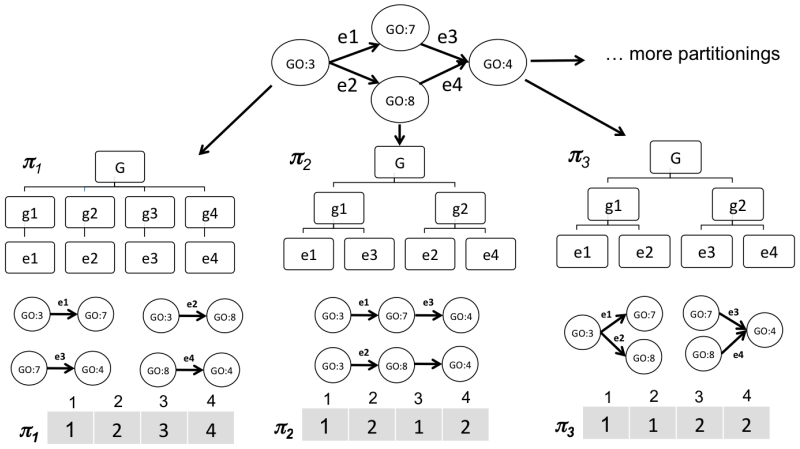


Figure 4.2: An example graph is partitioned into smaller subgraphs using partitioning functions p1, p2 and p3. The vector representation of each partitioning is presented under each of the three example partitionings. For instance, partitioning p3 assigns edges 1, 2 to subgraph 1 and edges 3, 4 to subgraph 2. Additional possible partitionings are not shown.

quality reflects how good is a partitioning in identifying key components of that system (pathway in this case).

The search for best partitionings therefore required a scoring function that could be used to assign high score to a partitioning that highlights the most likely patterns. For a pathway graph (G) of a disease class C and a partitioning π, $P(G, \pi|C)$ was defined as the probability of observing a pathway graph G and a partitioning given a disease class C. The value of $P(G, \pi|C)$ depended on how good that partitioning highlighted key subgraph patterns. Recall that H_π was defined as the set of subgraphs according to a partitioning function π of graph G: $H_\pi = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \pi(e) = i\}$

The graph partitioning probability $P(G, \pi|C)$ was then computed as a function of the set of subgraphs $g \in H_\pi$

$$P(G, \pi|C) = P(g_1, g_2, ..., g_n|C) \tag{4.1}$$

where $g_1, g_2, ..., g_n \in H_\pi$. Assuming subgraphs resulting from a partitioning function were conditionally independent, $P(G, \pi|C)$ was written as

$$P(G, \pi|C) = \prod_{g \in H_\pi} P(g|C) \tag{4.2}$$

The probability $P(g|C)$ represented the degree to which a subgraph g was a fingerprint of a disease class C. For the purpose of probability estimation, counting the number of instances of a given subgraph in partitionings of all graphs in a direct way was deemed impractical. This was because deciding whether two subgraphs were the same would require a test of subgraph isomorphism (Read and Corneil 1977). An indirect method was thus used to approximate subgraph matching by representing each subgraph with a set of maximal paths connecting its nodes. A maximal path was defined as a path that could not be extended by adding nodes to either end. The probability $P(g|C)$ could then be expressed in terms of probabilities of maximal paths given a class C. GO- annotated maximal paths inside the subgraphs were used to approximate representation of subgraphs, and thus avoid subgraph isomorphism test. Each maximal path represented a sequence of GO annotations of nodes that lay in that maximal path. In the case where a node had more than one GO annotation, multiple maximal paths were generated so that each maximal path had only one GO annotation per node. Then, $P(g|C)$ was calculated approximately as:

$$P(g|C) = \prod_{a \in g} P(a|C) \tag{4.3}$$

where a denotes a GO-annotated maximal path that connected a subset of nodes inside subgraph g. Using Equations 4.2-4.3, the likelihood of a partitioning and a graph instance given a disease class label was written as

$$P(G, \pi|C) = \prod_{g \in H_\pi} \prod_{a \in g} P(a|C) \tag{4.4}$$

Thus, Equation 4.4 represented a scoring function that was used in the search for best partitionings that highlighted disease fingerprints within pathway graphs. The problem was then that the probability distribution of maximal paths $P(a|C)$ did not exist a priori and needed to be estimated while searching for best partitionings. To solve this problem, an iterative training algorithm was used (described in the next section).

Parameter Estimation

The proposed model had a set of parameters $\theta = \{P(a|C)\}$ composing entries of the conditional probability table of maximal paths. The parameter set θ needed to be estimated in order to score graph partitionings. The Expectation Maximization (EM) (Dempster et al. 1977) algorithm was used to estimate model parameters according to Equations 4.2-4.4 while identifying the set of best partitionings for each graph in the pathway dataset. Initially, a set of random partitionings was generated and maximal paths within these partitionings were collected and an initial distribution for $P(a|C)$ was created. The parameter estimation process for this study had two basic steps. The first step was to search for highly scoring partitionings using the most recent probability table $P(a|C)$ obtained in the previous iteration of EM. Then, counts of maximal paths were collected from subgraphs of the set of best partitionings obtained. Collected counts were then normalized to produce a conditional probability model. During searching

and scoring of partitionings, a small probability value was used as a value of $P(a|C)$ in the case where a maximal path had not been added yet to the probability table. The EM algorithm was run for four iterations in this study. Additional details of the mathematical model and EM parameter estimation procedure are presented in Appendix A.

The EM algorithm had two outputs: the conditional probability table $P(a|C)$ and the set of best partitionings of each pathway in the dataset. Disease fingerprints were extracted from best partitionings of pathways. Using the model described above, the search for disease fingerprints not only depended on an individual score of a subgraph (according to Equation 4.3), but also based on the contributions of other subgraphs in the quality of a graph partitionings (according to Equation 4.4). To test the model, a graph classifier was built to classify pathways using probability table $P(a|C)$ that was estimated during EM run. This classification task served as benchmarking of the proposed model.

Benchmarking of the Fingerprint Mining Method

The efficacy of the structural pattern analysis method was demonstrated by implementing a graph classifier for disease pathways that utilized the conditional probability model estimated during model training. Given a test set of graphs, the task of the classifier was to assign the most likely disease class to each graph in a test set.

Classification of pathways

This classification task was modeled mathematically by finding the value for C that maximized $P(C|G)$, which represented the probability that C is a disease class of pathway G. Using Bayes theorem:

$$P(C|G) = \frac{P(C)P(G|C)}{P(G)} \tag{4.5}$$

where $P(C)$ quantified a priori knowledge about class label distribution, $P(G|C)$ was defined as the conditional probability of observing graph G given that its class label was C, and $P(G)$ was the probability distribution of graphs. The choice of class label did not depend on $P(G)$. Therefore, Equation 4.5 was expressed as

$$P(C|G) \propto P(C)P(G|C) \tag{4.6}$$

Modeling $P(G|C)$ directly would have required counting number of instances of a graph G. This approach had a practical challenge: Because each pathway was represented only once in the dataset, $P(G|C)$ would have followed a uniform distribution with probability equal to $1/(number of pathways of class C)$, and that would not have helped the statistical inference process. An alternative approach to model $P(G|C)$ that was used in this study was to incorporate the subgraph patterns in G according to the set of partitioning. Subgraphs tended to be more frequent in the dataset than their super graphs. Since one cannot be sure about which partitioning is the best, $P(G|C)$ was expressed in this study as the sum of best partitionings for graph G,

$$P(G|C) = \sum_{\pi} \prod_{g \in H_\pi} \prod_{a \in g} P(a|C) \tag{4.7}$$

Hence, having a prior distribution $P(C)$ and a conditional probability $P(G|C)$ that was calculated using partitionings and maximal paths conditional probability distribution, the classification problem was to find a class label C^* that maximized Equation 4.6:

$$C^* = argmax_C P(C)P(G|C) \tag{4.8}$$

During classification process, the search for a set of best partitionings was performed for each test graph instance (in the same way it was performed during probability estimation). The classification process started with setting a hypothesized class label C_0 for a test graph. Then, a search for the best partitioning set started with class label of test graph fixed to C_0. Equation 4.6 was used to evaluate $P(C_0|G)$. Then, another class label was used as a value for C_0, and a new set of partitionings was searched for and Equation 4.6 used to calculate $P(C_0|G)$. The class label that achieved the highest score was reported as classifier output. After benchmarking the graph structural pattern analysis method, the next module used the identified GO functionality patterns to predict subsystems in the GO-annotated interaction networks for a set of 14 species. This pattern matching module had two components, which are described in the following two subsections.

4.3.3 Functional Annotation and Indexing of Gene/Protein Interaction Networks

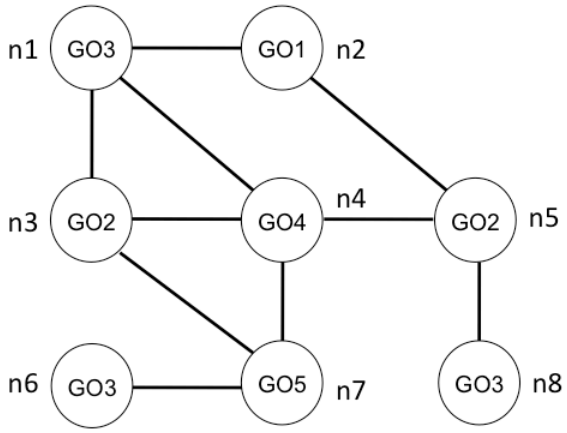
For each species, a network of genetic and protein interactions was constructed by importing interactions from two sources: BioGRID (Stark et al. 2006) and STRING (Szklarczyk et al. 2011). BioGRID data contains curated interactions from high throughput datasets and individual focused studies. In this study, only interactions within the same species were included. For some species analyzed in this study, the number of interactions was limited in BioGRID. To increase coverage of a species interaction network, more

interactions were imported from STRING database (version 9.0). STRING provides information about experimental and predicted interactions. Seven sources of information about a given interaction are used in STRING, including: genome context methods, gene co-expression, text mining, as well as associations known from other database resources such as BioCyc (Caspi et al. 2008) and PDB (Berman et al. 2000). An interaction in STRING database has a combined score that is computed using evidence scores from each data source. In this study, for data imported from STRING database, only interactions with combined score greater than or equal to 70% confidence were used in the construction of networks. Since fingerprints consisted of only GO terms (i.e., not gene/protein names) interaction networks of each species were GO- annotated in order to be suitable to match disease fingerprints learned from GO-annotated disease pathways. Nodes of interaction networks were annotated with molecular function annotations from the AmiGO Gene Ontology database (Carbon et al. 2009).

In this study, an interaction network of a given species could have had as many as 12,000 nodes (genes/proteins) and as many as 50,000 edges (interactions). Network indices were created for these large networks to enable efficient sub-network searches.

An index of an interaction network was built by generating a hash table with keys composed of ordered pairs of GO terms with first component being the node identifier of the node being indexed and second component denoting one of its neighbors. Values in the index table are identifiers of nodes with label equal to the first component of the ordered pair key. A value of a given key can be a single node identifier or a set of node identifiers. The index table was constructed by traversing every node in a given interaction network

and examining its neighboring nodes.



Index Table	
Key	Value
(GO1,GO2)	{n2}
(GO1,GO3)	{n2}
(GO2,GO1)	{n5}
(GO2,GO3)	{n3,n5}
(GO2,GO4)	{n3,n5}
(GO2,GO5)	{n3}
(GO3,GO1)	{n1}
(GO3,GO2)	{n1,n8}
(GO3,GO4)	{n1}
(GO3,GO5)	{n6}
(GO4,GO2)	{n4}
(GO4,GO3)	{n4}
(GO4,GO5)	{n4}
(GO5,GO2)	{n7}
...	

Figure 4.3: An example interaction network and an index with keys of GO annotations.

Figure 4.3 shows an example of GO-annotated interaction network and its index. In this example, suppose node $n1$ is to be indexed. Its neighbor nodes are $\{n2, n3, n4\}$. For the pair $(n1, n2)$ the corresponding annotation pair is $(GO3, GO1)$. A key of $(GO3, GO1)$ is inserted into the index with value $\{n1\}$. Similarly, the key $(GO3, GO2)$is inserted with value $\{n1\}$. In case a key already exists, values are appended to ones that already exist. For instance, when indexing node $n8$ that is annotated with $GO3$, a key $(GO3, GO2)$ already exists in the table with value being $\{n1\}$. Therefore, the value set is updated by

adding element $n8$ and the final key:value pair will be $(GO3, GO2) : \{n1, n8\}$.

4.3.4 Predicting Novel Subsystems using Disease Fingerprints

Disease fingerprints were identified using the method described in section 4.2.2 were matched to the GO annotated interaction networks (with interactions imported from BioGRID and STRING databases) using a similarity search algorithm. This algorithm used a network index to find subnetworks that matched an input disease fingerprint. Given a query subgraph and using the network index, the algorithm went through three steps.

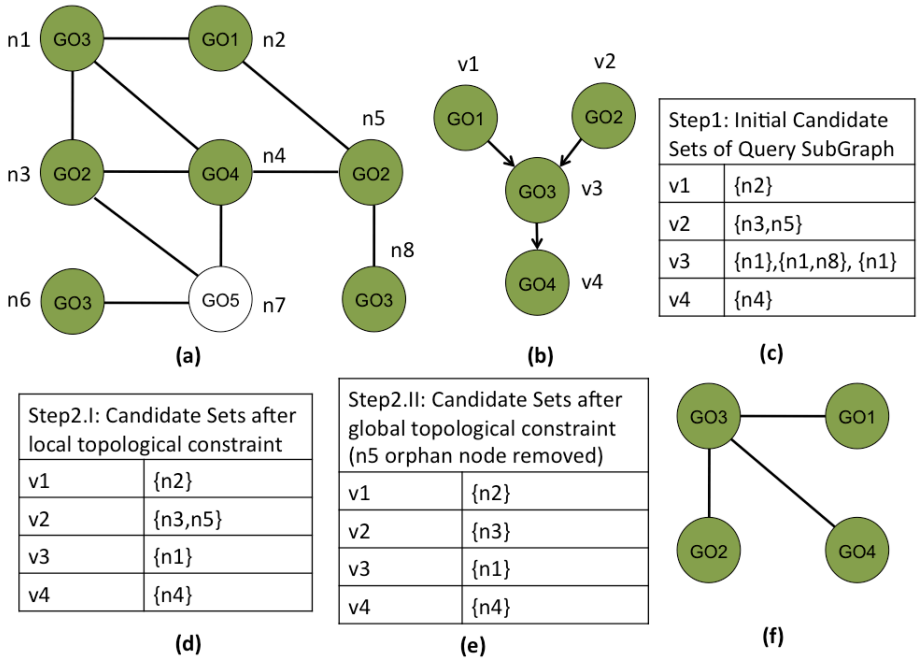


Figure 4.4: The process of matching a query subgraph (GO-annotated nodes) (b) to an interaction network (a). The three steps process start with generating initial candidate set of network nodes that match the GO terms of query subgraph nodes (c). The second step ([d] and [e]) refines candidate sets by removing network nodes that do not meet topological constraints. The last step is to generate an output subnetwork as answer to a query subgraph (f).

In the first step, an initial set of matched node identifiers (the candidate matching set) was retrieved for every node in the query subgraph. This was performed by using GO terms of nodes of each edge in the fingerprint subgraph to search the index. The following is an example to illustrate the pattern matching process (see Figure 4.4). Let v be a node in a query subgraph. For simplicity of demonstration, presume that each node has only one GO annotation. For every node u with an edge leading to v, an ordered pair of GO terms u_t and v_t was used as a key to lookup the network index. As a result, sets of node identifiers values of the corresponding key were retrieved from the table. For example, as shown in Figure 4.4.c, three sets of network node identifiers that matched query node v_3 (one set per neighbor). The first set resulted from the edge (v_3, v_1), with a key consisting of $(GO3, GO1)$. By looking the value up in the index table, the retrieved value was the set $\{n_1\}$. The second candidate set for query node v_3 resulted from the edge (v_3, v_2), with a key consisting of $(GO3, GO2)$. By looking this value up in the index table, the retrieved value of this key was the set $\{n_1, n_8\}$. Similarly, the third candidate set for query node v_3 resulted from the edge (v_3, v_4), with a key consisting of $(GO3, GO4)$, with $\{n_1\}$ as third candidate set for query node v_3 (see Figure 4.4.c). The process was repeated for every node in the query subgraph.

The second step was to examine candidate node identifier sets for each query node and to check topological constraints. A member in the candidate node set conforms to topological constraints if it has link to a member of other candidate node sets of neighbor nodes. Topological constraints were checked first by performing set intersection operation of all candidates sets of a given query node. For example, the final candidate set for query node v_3 was $\{n_1, n_8\} \cap \{n_1\} \cap \{n_1\} = \{n_1\}$. If the set intersection operation returned

empty set, then it would mean failure to match the query subgraph to any subnetwork in the interaction network, and hence the search was stopped. Node identifiers in the candidate set were then removed if they did not have any links to any node in candidate sets of other neighboring query nodes. For example, the node identifier n_5 in the candidate sets of v_2 (see Figure 4.4.d-e) was removed from that candidate set, because it was not connected to one item from candidate set of v_3 (n_5 was supposed to be connected to n_1 according to the query subgraph structure, but in the interaction network there was no link between node n_1 and node n_5). This step was repeated until all network node identifiers in query subgraph candidate sets satisfied topological constraints.

The third and final step was the generation of a set of sub-networks from candidate nodes sets of every query subgraph node. If there was only one node identifier for each candidate sets of query nodes, then it meant there was only one subnetwork that matched the input query subgraph. Otherwise, multiple subnetworks were returned as a matched set of the query subgraph. Details of the subgraph matching method are provided in Algorithm A.1 of Appendix A. The output of this algorithm was a set of subnetworks that served as candidate subsystems that partially or completely matched known pathways available in literature.

4.3.5 Scoring Candidate Subsystems

For each disease category, fingerprints were used to find subsystems in the interaction network for each of the 14 species. To evaluate these candidate subsystems, a set of reference pathways was used to determine the degree of matching between predicted subsystems and known pathways. A candidate subsystem was considered as being predicted correctly if

70% or more of its genes/proteins were found in a known pathway in a reference dataset. The Wikipathways database (Pico et al. 2008) was used as reference dataset. As recommended on the WikiPathways download page, only the analysis collection pathways were used for evaluation. Schizosaccharomyces pombe, Escherichia coli and Sus Scrofa had no WikiPathways analysis collection data. Also, since the pathways of Saccharomyces c. S288c and Arabidopsis thaliana in WikiPathways data were mainly metabolic pathways, they were not used to evaluate the predicted pathways. Predicted pathways of Escherichia coli and Saccharomyces c. S288c were matched to reference pathways from BioCyc. Reference pathways for Arabidopsis thaliana were downloaded from AraPath database (Lai et al. 2012). A further detailed evaluation for each species was reported for each disease of cancer and infectious disease classes in the design set.

4.4 Results

Evaluation of the developed approach was done in two steps. The first step was to measure the performance of the proposed mathematical model for structural pattern analysis as a function of the accuracy of a graph classifier. The second step was to evaluate the predicted subsystems that were discovered by the subgraph matching algorithm using a set of fingerprints for each disease class, and then comparing the discovered subsystems to known pathways published in the literature.

4.4.1 Datasets

The experiments were performed on disease pathways downloaded from KEGG pathway database (in September 2012). The KEGG disease pathways consisted of 63 pathways

distributed over seven disease classes. This dataset is summarized in Table 4.1. The gene/protein nodes of the pathway dataset were annotated with GO molecular function terms imported from HPRD database. Interaction networks for 14 species were downloaded from the BioGRID and STRING databases (in October 2012). All networks were annotated with molecular function annotations from AmiGO database. The GO molecular function hierarchy included a total of 10,286 GO concepts (as of July 2012). To determine the overall accuracy of the approach presented here, the candidate subsystems identified in the 14 interaction networks were compared to published pathways in WikiPathways and BioCyc databases.

Table 4.1: KEGG disease pathway categories.

Disease category	**Number of instances**
Cancer	17
Infectious disease	22
Substance Dependence	5
Neurodegenerative	5
Immune disease	7
Cardiovascular disease	4
Metabolic disease	3

4.4.2 Benchmarking of Structural Pattern Analysis Model

Given the set of 63 disease pathways analyzed for this study from KEGG, two binary classifiers were developed: (1) a cancer classifier and (2) an infectious diseases classifier. Cancer and infectious diseases had the largest number of instances in the design dataset (17 cancer pathways and 22 infectious diseases pathways, respectively). Two modified datasets were created: (1) a cancer dataset where graph instances were labeled as either associated

with cancer (positive case) or not associated with cancer (negative case; for this cancer classifier dataset, all non-cancer pathways such as infectious diseases, immune diseases, and neurodegenerative pathways were labeled negative); and, (2) an infectious disease dataset where graph instances were labeled as either associated with infectious disease (positive case) or not associated with infectious disease (negative case). A three-fold cross validation experiment was performed. The results of classification performance in terms of the geometric average of sensitivity and specificity are shown in Table 4.2. An overall accuracy of 86% was achieved.

Table 4.2: Average classification accuracy.

Disease category	**Average specificity**	**Average sensitivity**	**Overall average accuracy**
Cancer	0.75	1.0	0.87
Infectious	0.86	0.82	0.84

4.4.3 Assessment of Organisms as Molecular Models

Assessment of organisms as molecular models was performed by matching disease fingerprints identified in disease pathways to interaction networks for 14 organisms to find candidate subsystems. Evaluation results of predicted candidate subsystems for the 14 species analyzed in this study are shown in Table 3, including the proportions of known reference pathways that were recovered by the pathway prediction method. For instance, 61% of Bos taurus pathways in Wikipathways were recovered. Table 4.3 contains the number of interactions imported from BioGRID and STRING databases. As shown in Table 4.3, interaction networks of Bos taurus, Mus musculus, Rattus norvegicus, Danio

rerio, and Escherichia coli achieved the top five correctly predicted pathways among the species included in this study. The number of individual and summative interactions shown in Table 4.3 demonstrates the impact of importing data from STRING database with regard to size of interaction network for the top five species in terms of proportion of predicted subsystems nearly matching reference pathways dataset. Some species had no data in the reference set of pathways imported from WikiPathways. In particular, Schizosaccharomyces pombe and Sus scrofa had predicted subsystems that could not be evaluated. For STRING data, zero imported interactions means that the specified threshold of evidence score was not reached or there were already enough interactions from BioGRID (e.g., Saccharomyces c. S288c has 234,870 BioGRID interactions and thus no additional STRING interactions were imported). Sus scrofa did not have any reference pathways in WikiPathways, so no prediction accuracy could be reported.

Tables 4.4 and 4.5 show detailed performance of each species with respect to individual cancer and infectious diseases. Each column in Tables 4.4 and 4.5 shows the proportion of correctly predicted pathways for each of the 14 species analyzed in this study based on matching fingerprints between disease category specific and species interaction networks. The numbers of correctly predicted pathways per species were normalized to give proportions such that each species covered a set of fingerprints for a disease. As examples of correctly predicted pathways using cancer disease fingerprints, the proposed method successfully recovered 11 out of 16 genes in the androgen signaling pathway (PW:0000564), five out of six genes of the altered canonical Wnt signaling pathway (PW:0000599) and five out of six genes in tamoxifen pharmacodynamics pathway (PW:0000839) from the published Rat Genome Database (RGD) (Dwinell et al. 2009).

4.5 Discussion

In silico identification of potential model organisms may be a cost effective first step in the study of human diseases. By annotating genetic pathways with GO terms, subgraph patterns in genetic pathways can acquire greater generalization capability. This generalization allows for matching with an organisms interaction network that was also annotated using GO terms. The degree to which an interaction network of a given model organism covered subgraph patterns of disease pathways was hypothesized to be a measure of the suitability of this model organism to study biological processes related to human diseases. A significant proportion of the interactions (genetic and physical) used in network construction were predicted interactions (e.g., inferred by genome context methods or text mining). This allowed for the evaluation of organisms as potential disease models even with limited curated interaction data.

4.5.1 Main Findings

The statistics in Tables 4.3 - 4.5 show the range of disease model suitability for the 14 analyzed organisms in terms of pathways prediction accuracy. The interaction networks of *Arabidopsis thaliana* (mouse-ear cress; a plant) and *Escherichia coli* (a bacterium) performed better than hose of gallus(chicken), *Canis lupus familiaris* (dog), or *Bos taurus* (cow) in predicting pathways using disease fingerprints of colorectal as well as thyroid cancer (see Table 4.4). Additionally, interaction networks of *Sacchromyces cerevisiae* (Bakers yeast) performed better than *Mus musculus* (mouse) or *Rattus norvegicus* (rat) in predicting pathways using *Eppstein- Barr virus* disease fingerprints (see Table 4.5). These types of findings are supported by McGary, et al., where organisms such as *Sacchromyces*

cerevisiae and *Caenorhabditis elegans* (in contrast to *Mus musculus* or *Rattus norvegicus*) were described as putative model organisms for human diseases (McGary et al. 2010).

This study was different from the approach of McGary, et al. in the way that it depends on network structure of genetic pathways as well as Gene Ontology annotations. The work of McGary, et al. was based on overlapping sets of orthologous genes, and a mathematical formulation based on these sets was used to find model organisms. The work of McGary et al. was based on molecular sequence information, without using network analysis to rank model organisms based on predicted subsystems (although McGary, et al. studied connectivity and modularity of the subsystems they discovered in cellular networks of candidate organisms, but that was a further analysis step of the results and was not a core part of their described method).

Based on the results shown in Tables 4.4 and 4.5, it was also observed that performance of *Mus musculus* and *Rattus norvegicus* models was greatly different in the case of some cancer diseases (e.g., *Renal cell carcinoma* and *Melanoma*) and infectious diseases (e.g., *Pertussis* and *Epstein-Barr Virus*). These results suggest that it may be worth exploring *Danio rerio* (for *Renal cell carcinoma*, *Melanoma*, or *Pertussis*) or *Saccharomyces cerevisiae* (for *Epstein-Barr Virus*) as better disease models for certain diseases. To further support this finding, recent studies have proposed *Danio rerio* as a potential model organism for cancer (Stoletov and Klemke 2008, Stern and Zon 2003, Feitsma and Cuppen 2008), infectious and immune diseases (Sullivan and Kim 2008), and in vivo drug discovery (Zon and Peterson 2005). Furthermore, some genes of *Saccharomyces cerevisiae* have shown similarity to *Epstein-Barr virus* DNA polymerase and be orthologous to human

genes associated with *Epstein-Barr virus* (Morrison et al. 1989, Dheekollu and Lieberman 2011). However, it is important to note that the plausibility of alternative model organisms might also require the consideration of other features such as phenotypic properties of these specific diseases (e.g., do the organisms exhibit an observable disease state phenotype that is alterable?) as well as other practical considerations (e.g., availability of valid wild-types or appropriate inbred species).

4.5.2 Choice of Data Resources and Annotation Scheme

Combining micro-level, molecular function annotations of gene/protein nodes together with information about semantics inherited in a graph structure can be a powerful approach to derive new findings of relevance to biomedicine. Node annotations might not be restricted to molecular function annotations of GO. Genes/proteins in pathways and interaction networks with disease- specific annotation could be augmented from a variety of knowledge sources. For example, it may be possible to leverage biobanking and phenotypic information from Electronic Health Records (EHR) (Jensen et al. 2012) and clinical data resources to annotate disease genes/proteins. Indeed, we are currently exploring the potential to do this in the future, with the goal to develop an EHR knowledge-enriched model to study disease genes/proteins in the context of real clinical scenarios.

While GO annotations can be found in gene ontology annotations (GOA) files of the Gene Ontology database, HPRD was chosen as a source of GO annotations because it is a manually curated resource and GO-compatible database. HPRD initially started with data from the Online Mendelian Inheritance in Man (OMIM) database (Hamosh et al. 2005)

that focused on disease related genes (Peri et al. 2003). This level of curation met the scope of this study to learn knowledge from disease- related genetic pathways.

This study only made use of GO molecular function terms. GO biological process terms are more diverse (and more specific) in characterizing genes/proteins than molecular function terms (there are nearly 2.5 times more biological process terms than molecular function terms). For the purposes of this study, molecular function terms were able to increase the model generalization (extracted patterns can be matched to GO-annotated interaction networks), thus not requiring additional biological process terms. Even though the GO biological process terms were not used in the model, the KEGG edge annotations (e.g., general process type such as PPrel and specific relation types such as activation, expression and inhibition) do capture semantics of the biological process that involved two genes/proteins.

Using a major gene/protein interaction database such as BioGRID, which provides a high number of unique interactions among other major databases (Lehne and Schlitt 2009), can be a limiting factor for predicting subsystems in many species due to low number interactions for some species in BioGRID database. The use of gene/protein interactions drawn from meta-databases such as STRING enhanced the ability to recover known subsystems by increasing the size of interaction networks. The number of interactions (per species) imported from BioGRID and STRING databases highlights the importance of aggregating evidence information about interactions from large number of sources. For instance, the interaction network of *Escherichia coli* had only four interactions in BioGRID database. About 50,000 interactions regarding *Escherichia coli* imported

from STRING enabled the prediction of 23% of Wikipathways reference pathways of *Escherichia coli*. For Danio rerio, the interaction network had only 112 interactions imported from BioGRID. Importing 47,029 interactions from STRING allowed for 18% prediction accuracy for cancer diseases class and 12% prediction accuracy of infectious diseases class. The majority of interactions imported from STRING regarding *Escherichia coli* and *Danio rerio* were largely supported by evidence scores from predicted interactions (e.g., genome context and text mining).

The contribution of multiple methods for interaction prediction can be demonstrated by the case of *Danio rerio* and *Escherichia coli* interaction networks constructed using interactions imported from STRING. As shown in Figures 4.5 and 4.6, about 55% of the interaction network constructed for the *Danio rerio* and about 80% of the interaction network constructed for the *Escherichia coli* were derived from evidence from experimental, gene expression, text mining, and gene neighborhood methods that collectively increased the overall evidence score above 70%. As has been done by others (e.g., Ferrer, et al. (Ferrer et al. 2011) used threshold of 50% for an adjusted rand index for determining the correctness of a pathway), a threshold of 70% was mainly chosen to imply that more than two thirds of the genes/proteins in a pathway are found. However, if the *Danio rerio* and *Escherichia coli* networks were constructed only from data imported from major databases, the networks would respectively be 45% and 20% of their potential size. Table 4.6 shows statistics about the STRING interactions used in the construction of the *Danio rerio* network. While 98% of *Danio rerio* network links had non-zero scores for partial evidence derived from other databases, 55% these partial evidence scores would not pass the 70% threshold and hence the *Danio rerio* networks would be 45% of its size.

Partial evidence from experimental, gene expression, text mining, and gene neighborhood methods thus boosted the size of *Danio rerio* network. The results shown in Table 4.4 also suggest that, for some species, very few known interactions (118 as in the *Danio rerio* dataset) were available in BioGRID database. Including interactions from STRING (mostly predicted interactions) allowed for a wider coverage of the interaction network. The overall impact of including multiple sources resulted in an improvement of overall prediction accuracy for 18% of subsystems discovered by cancer fingerprints and 12% for infectious disease fingerprints.

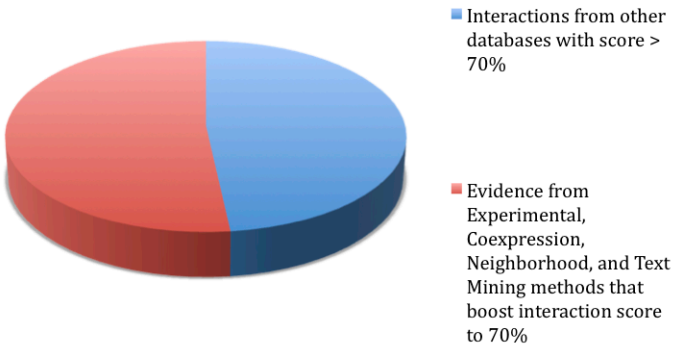


Figure 4.5: Contribution of methods used to predict interactions for the construction of interaction network of Danio rerio.

4.5.3 Summary of Study Contributions

There are four major contributions of the methodology developed in this study for evaluating potential model organisms. First, it was shown that a model-based method could be used to search and extract functional structural patterns (disease fingerprints) in disease pathway graphs. Second, a subgraph pattern matching algorithm, supported by a simple

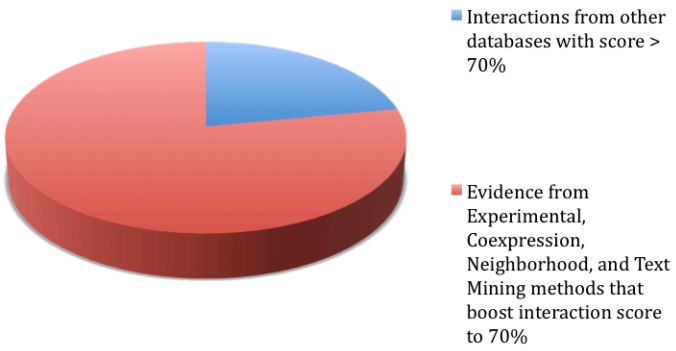


Figure 4.6: Contribution of methods used to predict interactions for the construction of interaction network of Escherichia coli.

and memory-efficient indexing method was shown to be useful for identifying subsystems in interaction networks using disease fingerprints. Third, this work leveraged rich knowledge sources (KEGG pathways, BioGRID and STRING interactions databases that could be annotated with GO) together with computational mining methods to infer potentially new knowledge (e.g., novel subsystems of disease). The fourth, and perhaps most significant, way that the methodology presented here is different from previous studies is that the assessment of disease model potential was achieved at both the unit level (by considering molecular function) and system level (by considering graph structure patterns in pathways). Thus, this approach is different from related studies that used gene ortholog sets as the basis to assess how an organism was suitable as a model (e.g., most recently by McGary, et al. (McGary et al. 2010)). The method used in this study complements these types of approaches in two major ways: (1) the way gene molecular function is used to represent similarity of genes in different organisms and (2) pathways are predicted using system-level graph-based methods.

4.5.4 Study Limitations

The methods presented here have a number of limitations related to decisions about the computational methods, the data resources, and the assumptions made in this study. The EM algorithm that was used for parameter estimation (see Appendix A) is known for not guaranteeing optimum solutions. Graphs in the KEGG pathway datasets were assumed to be independent and identically distributed data. While it is hard to confirm that a given pair of pathways sharing a set of genes/proteins is totally independent, assuming independence of graphs items was for the purpose of statistical analysis and to make the computation of the model more tractable. There are a number of alternative resources that might have been used, including Reactome pathways and molecular networks (Joshi-Tope et al. 2005), species-specific databases such as Rat Genome Database (RGD) (Dwinell et al. 2009) and WormBase(Harris et al. 2010).

Some limitations are inherent in the datasets chosen for this study and could have had an impact on the results produced. For instance, significant proportions of the interactions in STRING database are predicted interactions and thus there is always a possibility of errors about predicting two genes/proteins being genetically or physically interacting. There might be gene set overlap between pathway data from Wikipathways, BioCyc, and AraPath databases with the KEGG human pathways that were used as design dataset. However, this did not have a significant effect on quality of evaluation procedure for two reasons. First, the method described in this study did not use any sequence similarity or homology-based technique to predict pathways similar to those of humans in other species. Second, the methodology used in this study relied on the molecular function of genes, not the genes themselves and therefore, genes in predicted pathways did not necessarily have

to be sequence-based homologs of human genes.

Evaluating candidate model organisms at the molecular level is only one facet for determining the viability of a possible model organism. Other factors, such as cost and controllability in a lab environment, also need to be considered. This study aimed to utilize already available resources about potential model organism for systematic evaluation, without particular consideration of cost or controllability. Nonetheless, the use of the approach described in this study may be one factor that can be combined with cost and controllability factors to help guide future research on human diseases.

4.6 Conclusion

This study proposed a method for the evaluation of species as models to study human diseases. Disease-related genetic pathways were functionally and structurally analyzed to uncover characteristic subgraph patterns. These patterns were then matched to molecular interaction networks for 14 potential model organisms. The adequacy of a given species as a potential disease model was hypothesized to be related to the degree to which interaction networks cover disease patterns. The finding that proportions of correctly predicted subsystems in *Danio rerio* (Zebrafish) and *Saccharomyces cerevisiae* (Baker's yeast) interaction networks were higher than those of two common model organisms *Mus musculus* (Mouse) and *Rattus norvegicus* (Rat) suggests there might be unobvious molecular networks in alternative model organisms that might be relevant to study disease-related processes. The findings of this study suggest that a network, system-level approach can be an effective means to find such unobvious networks. The promising results of this study suggest that the disease fingerprint approach may be used to analyze pathways across multiple species

and may thus be used to identify model organisms for the study of human disease related processes.

4.7 References

Aitman, T. J., C. Boone, G. A. Churchill, M. O. Hengartner, T. F. C. Mackay, and D. L. Stemple (2011). The future of model organisms in human disease research. *12*.

Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, and J. T. Eppig (2000). Gene ontology: tool for the unification of biology. *25*.

Bebek, G. and J. Yang (2007). Pathfinder: mining signal transduction pathway segments from protein-protein interaction networks. *BMC bioinformatics 8*(1), 335.

Bedell, M. A., D. A. Largaespada, N. A. Jenkins, and N. G. Copeland (1997). Mouse models of human disease. part ii: recent progress and future directions. *Genes & development 11*(1), 11–43.

Berman, H., J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne (2000). The protein data bank. *Nucleic acids research 28*(1), 235–242.

Cakmak, A. and G. Ozsoyoglu (2007). Mining biological networks for unknown pathways. *Bioinformatics 23*(20), 2775.

Carbon, S., A. Ireland, C. Mungall, S. Shu, B. Marshall, and S. Lewis (2009). Amigo: online access to ontology and annotation data. *Bioinformatics 25*(2), 288–289.

Caspi, R., H. Foerster, C. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S. Rhee, A. Shearer, and C. Tissier (2008). The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic acids research 36*(suppl 1), D623–D631.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–38.

Dheekollu, J. and P. M. Lieberman (2011). The replisome pausing factor timeless is required for episomal maintenance of latent epstein-barr virus. *85*.

Dwinell, M., E. Worthey, M. Shimoyama, B. Bakir-Gungor, J. DePons, S. Laulederkind, T. Lowry, R. Nigram, V. Petri, and J. Smith (2009). The rat genome database 2009: variation, ontologies and pathways. *Nucleic acids research 37*(suppl 1), D744–D749.

Feitsma, H. and E. Cuppen (2008). Zebrafish as a cancer model. *6*.

Ferrer, L., A. G. Shearer, and P. D. Karp (2011). Discovering novel subsystems using comparative genomics. *27*.

Haldar, M., J. D. Hancock, C. M. Coffin, S. L. Lessnick, and M. R. Capecchi (2007). A conditional mouse model of synovial sarcoma: insights into a myogenic origin. *Cancer cell 11*(4), 375–388.

Haldar, M., R. L. Randall, and M. R. Capecchi (2008). Synovial sarcoma: From genetics to genetic-based animal modeling. *Clinical Orthopaedics and Related Research 466*, 2156–2167.

Hamosh, A., A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick (2005). Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic acids research 33*(suppl 1), D514–D517.

Harris, T. W., I. Antoshechkin, T. Bieri, D. Blasiar, J. Chan, W. J. Chen, N. De La Cruz, P. Davis, M. Duesbury, and R. Fang (2010). Wormbase: a comprehensive resource for nematode research. *Nucleic acids research 38*(suppl 1), D463–D467.

Jensen, P. B., L. J. Jensen, and S. Brunak (2012). Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics 13*(6), 395–405.

Joshi-Tope, G., M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, and L. Matthews (2005). Reactome: a knowledgebase of biological pathways. *33*.

Kanehisa, M., S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa (2010). Kegg for representation and analysis of molecular networks involving diseases and drugs. *38*.

Koteja, P., T. Garland, J. K. Sax, J. G. Swallow, and P. A. Carter (1999). Behaviour of house mice artificially selected for high levels of voluntary wheel running. *Animal behaviour 58*(6), 1307–1318.

Lai, L., A. Liberzon, J. Hennessey, G. Jiang, J. Qi, J. Mesirov, and X. Steven (2012). Arapath: a knowledgebase for pathway analysis in arabidopsis. *Bioinformatics 28*(17), 2291–2292.

Lehne, B. and T. Schlitt (2009). Protein-protein interaction databases: Keeping up with growing interactomes. *Human genomics 3*(3), 291–297.

McGary, K. L., T. J. Park, J. O. Woods, H. J. Cha, J. B. Wallingford, and E. M. Marcotte (2010). Systematic discovery of nonobvious human disease models through orthologous phenotypes. *Proceedings of the National Academy of Sciences 107*(14), 6544–6549.

Morrison, A., R. Christensen, J. Alley, A. Beck, E. Bernstine, J. Lemontt, and C. Lawrence (1989). Rev3, a saccharomyces cerevisiae gene whose function is required for induced mutagenesis, is predicted to encode a nonessential dna polymerase. *171*.

Nabhan, A. R. and I. N. Sarkar (2012). Mining disease fingerprints from within genetic pathways. In *AMIA Annual Symposium Proceedings*, Volume 2012, pp. 1320. American Medical Informatics Association.

Peri, S., J. D. Navarro, R. Amanchy, T. Z. Kristiansen, C. K. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, T. Gandhi, and M. Gronborg (2003). Development of human protein reference database as an initial platform for approaching systems biology in humans. *13*.

Pico, A., T. Kelder, M. Van Iersel, K. Hanspers, B. Conklin, and C. Evelo (2008). Wikipathways: pathway editing for the people. *PLoS biology 6*(7), e184.

Prasad, T., R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, and A. Venugopal (2009). Human protein reference database - 2009 update. *Nucleic acids research 37*(suppl 1), D767–D772.

Read, R. and D. Corneil (1977). The graph isomorphism disease. *Journal of Graph Theory 1*(4), 339–363.

Senf, A. and X.-w. Chen (2009). Identification of genes involved in the same pathways using a hidden markov model-based approach. *Bioinformatics 25*(22), 2945–2954.

Stark, C., B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers (2006). Biogrid: a general repository for interaction datasets. *Nucleic acids research 34*(suppl 1), D535–D539.

Stern, H. M. and L. I. Zon (2003). Cancer genetics and drug discovery in the zebrafish. *Nature Reviews Cancer 3*(7), 533–539.

Stoletov, K. and R. Klemke (2008). Catch of the day: zebrafish as a human cancer model. *Oncogene 27*(33), 4509–4520.

Sullivan, C. and C. H. Kim (2008). Zebrafish as a model for infectious disease and immune function. *25*.

Szklarczyk, D., A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, and P. Bork (2011). The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *39*.

Thomas, M. A., L. Yang, B. J. Carter, and R. D. Klaper (2011). Gene set enrichment analysis of microarray data from pimephales promelas (rafinesque), a non-mammalian model organism. *12*.

Tian, L., S. A. Greenberg, S. W. Kong, J. Altschuler, I. S. Kohane, and P. J. Park (2005). Discovering statistically significant pathways in expression profiling studies. *102*.

Yan, X. and J. Han (2002). gspan: Graph-based substructure pattern mining. IEEE.

Zon, L. I. and R. T. Peterson (2005). In vivo drug discovery in the zebrafish. *4*.

Table 4.3: Number of interactions and proportions of predicted pathways that correctly matched reference pathways for a given species.

		Number of interactions			Proportions of correct predicted	
NCBI Taxon ID	**Species name**	BioGRID	STRING	Total	Cancer	Infectious disease
3702	Arabidopsis thaliana	13,828	0	13,828	0.0419	0.055
4896	Schizosaccharomyces pombe	17,495	32,505	50,000		
6239	Caenorhabditis elegans	6998	0	6998	0	0.004
7227	Drosophila melanogaster	40,153	9848	50,001	0.006	0.012
7955	Danio rerio	112	47,029	47,141	0.171	0.144
9031	Gallus gallus	180	39,337	39,517	0.067	0.063
9598	Pan troglodytes	0	36,756	36,756	0.158	0.153
9615	Canis lupus familiaris	5	33,398	33,403	0.014	0.014
9823	Sus scrofa	1	12,831	12,832		
9913	Bos taurus	33	49,967	50,000	0.614	0.192
10090	Mus musculus	4729	45,271	50,000	0.452	0.506
10116	Rattus norvegicus	851	49,163	50,014	0.391	0.289
511145	Escherichia coli	4	49,996	50,000	0.236	0.037
559292	Saccharomyces c. S288c	234,870	0	234,870	0.023	0.012

Table 4.4: Detailed performance analysis of 14 Species on cancer diseases fingerprints. Entries are the proportions of correctly predicted pathways for each of the 14 species.

NCBI Taxon ID	**Species name**	Colorectal cancer (hsa05210)	Renal cell carcinoma (hsa05211)	Pancreatic cancer (hsa05212)	Endometrial cancer (hsa05213)	Glioma (hsa05214)	Thyroid cancer (hsa05216)	Basal cell carcinoma (hsa05217)	Melanoma (hsa05218)	Bladder cancer (hsa05219)	Chronic myeloid leukemia (hsa05220)	Acute myeloid leukemia (hsa05221)	Non-small cell lung cancer (hsa05223)
3702	Arabidopsis thaliana	0.018	0.005	0.001	0.004	0	0.014	0	0	0	0	0	0
4896	Schizosaccharomyces pombe	0	0	0	0	0	0	0	0	0	0	0	0
6239	Caenorhabditis elegans	0	0	0	0	0	0	0	0	0	0	0	0
7227	Drosophila melanogaster	0.004	0.001	0	0.002	0	0.001	0	0	0	0	0	0
7955	Danio rerio	0.04	0.459	0.059	0.226	0.07	0.08	0.015	0.449	0.215	0.228	0.077	0.098
9031	Gallus gallus	0.001	0.004	0	0.031	0	0.012	0	0	0	0	0	0
9598	Pan troglodytes	0.03	0.049	0.007	0.084	0.156	0.064	0	0.094	0	0.07	0.092	0.096
9615	Canis lupus familiaris	0.005	0.004	0	0.019	0	0.011	0	0	0	0.008	0.01	0.014
9823	Sus scrofa	0	0	0	0	0	0	0	0	0	0	0	0
9913	Bos taurus	0.002	0	0	0.008	0.001	0	0	0	0	0	0	0
10090	Mus musculus	0.669	0.25	0.395	0.28	0.344	0.416	0.859	0.384	0.439	0.296	0.314	0.351
10116	Rattus norvegicus	0.184	0.229	0.536	0.348	0.428	0.402	0.122	0.072	0.346	0.398	0.508	0.441
511145	Escherichia coli	0.022	0	0	0	0	0	0	0	0	0	0	0
559292	Saccharomyces c. S288c	0.025	0	0.002	0	0	0	0.005	0	0	0	0	0

Table 4.5: Detailed performance analysis of 14 Species on infectious diseases fingerprints. Entries are the proportions of correctly predicted pathways for each of the 14 species.

NCBI Taxon ID	Species name	Bacterial invasion epithelium (hsa05100)	Helicobacter pylori infection (hsa05120)	Escherichia coli infection (hsa05130)	Shigellosis (hsa05131)	Pertussis (hsa05133)	Legionellosis (hsa05134)	Leishmaniasis (hsa05140)	Chagas disease (hsa05142)	Toxoplasmosis (hsa05145)	Tuberculosis (hsa05152)	Hepatitis C (hsa05160)	Measles (hsa05162)	Influenza A (hsa05164)	HTLV-I (hsa05166)	Herpes simplex (hsa05168)	Epstein-Barr virus (hsa05169)
3702	Arabidopsis thaliana	0.005	0	0	0	0	0	0.006	0.004	0.001	0.002	0.022	0.003	0.021	0.002	0.009	0
4896	Schizosaccharomyces pombe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6239	Caenorhabditis elegans	0.001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7227	Drosophila melanogaster	0.008	0	0	0	0	0	0.002	0.001	0	0	0.002	0.001	0	0	0.001	0
7955	Danio rerio	0.019	0	0	0.068	0.348	1	0.248	0.136	0.012	0.067	0.161	0.103	0.046	0.065	0.05	0
9031	Gallus gallus	0.005	0	0	0.002	0.001	0	0.013	0.008	0.005	0.006	0.024	0.002	0.001	0.002	0	0
9598	Pan troglodytes	0.019	0	0	0.058	0.158	0	0.083	0.038	0.072	0.055	0.081	0.046	0.033	0.064	0.007	0
9615	Canis lupus familiaris	0.003	0	0	0.013	0	0	0.006	0.003	0.002	0.004	0.009	0.001	0.001	0.004	0.003	0
9823	Sus scrofa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9913	Bos taurus	0.003	0	0	0	0	0	0	0.001	0.007	0.001	0.001	0	0	0	0.001	0
10090	Mus musculus	0.693	1	1	0.638	0.211	0	0.346	0.617	0.624	0.66	0.39	0.596	0.65	0.588	0.793	0.171
10116	Rattus norvegicus	0.227	0	0	0.221	0.282	0	0.296	0.189	0.265	0.202	0.31	0.247	0.245	0.274	0.125	0
511145	Escherichia coli	0.004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
559292	Saccharomyces c. S288c	0.014	0	0	0	0	0	0	0.004	0.011	0.002	0	0.001	0.005	0	0.01	0.829

Table 4.6: Interactions of *Danio rerio* interaction network with detailed sources of evidence.

Evidence Method/Source	**Number of STRING links with non-zero score**	**Percentage of Network links with non-zero score**
Neighborhood	4891	9.7
Fusion	299	0.6
Cooccurence	2219	4.4
Coexpression	22,319	44
Experimental	11,547	23
Other databases	49,103	98
Text Mining	15,149	30

Chapter 5

Graph-based Mining in Biomedical Literature for Assessment of Disease Model Organisms

Nabhan, A. R. and I. N. Sarkar (2013). Graph-based Mining in Biomedical Literature for Assessment of Disease Model Organisms. In preparation.

5.1 Abstract

Motivation: The identification of potential model organisms to study disease phenomena is an important task in biomedicine. The potential to identify potential model organisms based on evidence re-ported in biomedical literature may complement more traditional genomic and proteomic approaches. Natural language processing (NLP) methods enable the analysis of large collections of biomedical literature, such as indexed by resources like MEDLINE. This study aimed to leverage NLP-based annotations of MEDLINE to support

a graph-based mining technique for the identification of potential dis-ease model organisms.

Results: A graph-based mining method was developed that lever-aged syntactic and semantic annotations provided by NLP-processing done by MetaMap, a publicly available tool from the US National Library of Medicine. The approach was used to find semantically equivalent graph patterns across citations that reported evidence about organisms. These semantically equivalent patterns were then used to develop a quantitative assessment of described organisms as potential disease models.

5.2 Introduction

Model organisms are often used to study the underlying mechanisms of disease, enabling the study of genetically modified populations with phenotypes or traits of interest. Mice and rats dominate as the chosen models to study human diseases, while there are a handful of other organisms used for particular conditions (e.g., fruit fly and zebrafish). Recent advancements in genome sequencing technologies alongside the generally improved ability to acquire and catalogue biological information has led to an increased amount of genomic and proteomic materials for organisms that may not have been previously studied extensively. The availability of such resources provides the opportunity to evaluate the potential of *in silico* methods to identify potential model organisms for supporting the study and understanding of disease.

A common goal of biomedical research is to identify genes and their functions as well as associated molecular mechanisms or pathways for various cellular processes in sequenced organisms. Comparative genomics methods have been used to infer the

function of genes using information about previously studied organisms (Sandelin et al. 2004, Wolf et al. 2001). This has led to an increasing number of reports describing newly discovered mechanisms that underlie various phenomena, including those associated with disease etiology (Glass et al. 2010, Whelan et al. 2010). Scientific knowledge may be represented by relationships between domain concepts present in literature (Bodenreider 2004). However, this knowledge cannot be easily accessed and summarized to address scientific questions because it is generally embedded in free text (Altman and Klein 2002, Rebholz-Schuhmann et al. 2012). Data mining methods, coupled with natural language processing (NLP), may thus provide a means to uncover important patterns that summarize information embedded in biomedical literature (Yoo et al. 2007). The availability of biomedical knowledge in large, freely available resources (e.g. as citations indexed by MEDLINE) provides an opportunity to leverage data mining methods to find potential disease model organisms.

This study aimed to develop an approach to mine biomedical literature of a given organism to uncover patterns about phenomena and processes (e.g. molecular interactions, phenotypic features, and experimental procedures) that can subsequently be used to characterize a specific disease that the organism may be a suitable model for studying. These methods applied a graph pattern analysis model to develop a graph-based representation of sentences in biomedical citations (MEDLINE) in order to uncover significant subgraph patterns. This model allowed for the incorporation of knowledge based annotations to address the data sparsity problem (that is particularly common with textual data) and to increase the generalization capability of subgraph patterns. Generalization of subgraph patterns in this context meant that vertices of patterns were annotated with concepts from ontologies that

enabled patterns to be matched to other graphs whose vertices were also annotated with the same ontological concepts.

5.3 Materials and Methods

The method developed for this study used an annotated biomedical literature citation corpus (MEDLINE) to generate a graph-based representation of sentences. A graph pattern mining method was developed to highlight key patterns within the annotated corpus. The similarity of patterns relative to citations associated with describe human disease patterns formed the basis to suggest potential disease models.

5.3.1 Annotated Text Corpus

In order to generate concept graphs that represented sentences in a given MEDLINE citation, concepts (vertices) and relationships (edges) had to be identified. The MetaMap software tool, developed by the US National Library of Medicine (NLM), provides morpho-syntactic and semantic annotations of natural language text (Bodenreider 2004, Browne et al. 2003). For a given sentence or utterance, MetaMap identifies words and their part-of-speech tags, phrase structures, in addition to terms/concepts that can be mapped to UMLS concepts. In this study, MetaMap annotations were used as input for the later described method to generate concept graph representations for sentences from MEDLINE. The annotations used for this study were based on the pre-computed MetaMap Machine Output (MMO) repository (2012 release), which was the result of processing of more than 20 million citations from the MEDLINE using MetaMap.

Citations with Medical Subject Headings (MeSH) descriptors (Lipscomb 2000) related to diseases or organisms groups that were the focus of this study were selected from the MMO output for subsequent pattern mining steps. In addition to these scope restrictions, citations were also filtered based on MeSH descriptors contained within the Phenomena and Processes [G] MeSH tree hierarchy.

Table 5.1: Sample rules for mapping syntactic structures to concept graphs.

Phrase Structure Pattern	**Graph Creation Rules**
modifier, modifier, head	$e_1(2, attr, 0), e_2(2, attr, 1)$
preposition, modifier, head	$e_1(0, rel, 2), e_2(2, attr, 1)$
pronoun, modifier ,modifier, head	$e_1(0, rel, 3), e_2(3, attr, 1), e_3(3, attr, 2)$
head, preposition, modifier	$e_1(0, rel, 1), e_2(1, rel, 2)$

5.3.2 Transformation of Syntactic and Semantic Structures into Concept Graphs

Syntactic structures in MMO annotations (e.g. part-of-speech tags, phrase structures, and UMLS concept mappings) were used to construct concept graphs. For instance, a syntactic annotation of a noun phrase can indicate the head of the noun phrase (main noun) and a set of modifiers (e.g. adjectives). This annotation helped identify concepts (head of the phrase and its modifiers) and relationships between concepts (has-attribute relationship between the head and its modifiers). In concept graphs, each vertex has a type (e.g. concept, action) and edges can be labeled with types (Leskovec et al. 2004). Each vertex has a set of factors (or features) that describe the vertex such as label (e.g. inhibitory), part-of-speech tag (e.g. adjective), and semantic type (e.g. molecular function). Labels of edges represent relations

between two vertices. For instance, edge labels can be attr indicating that source vertex (e.g. head or noun) has an attribute described by destination vertex (e.g. adjective).

A set of transformation rules was developed to transform MetaMap annotations of sentences into concept graphs. Each rule described a phrase structure pattern and had an action describing how the concept graph was to be generated. The action of a rule specified the generation of a set of tuples describing edges of the concept graph. For instance, a rule can be of the form: MODIFIER,HEAD => (1,attr,0). This rule consists of a pattern of a simple Noun Phrase with a head (noun) and a modifier (adjective). The action is to generate an edge with source vertex index 1 (referring to the second word in this noun phrase), and edge label attr, and destination vertex index 0 (referring to the first word in this noun phrase). All phrase structures (e.g. Noun Phrase, Preposition Phrase, and Verb Phrase) had a set of transformation rules that were used to generate phrase concept graphs.

Then, concept graphs were generated by the merging smaller concept graphs of sentences constituent phrases. A set of rules was used to generate sentence-level concept graphs by linking the head or main vertex in one concept graph of a phrase to the head vertex of a concept graph of another phrase. Sentence-level graph generation rules described how smaller phrase-level graphs were to be merged. As an example of sentence-level graph generation rule, NP,conj,NP => (head(0), conj, head(2)) describes a sentence with two noun phrases linked by a conjunction (e.g. and). The action of this rule is to create an edge between the head of first phrase and the head of the third phrase in the rule pattern, with an edge label conj. A total of 230 transformation rules were written to generate concept graphs. A sample of transformation rules is given in Table 5.1.

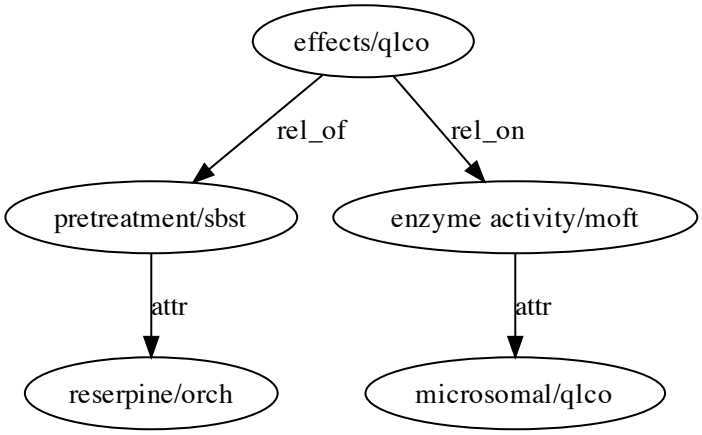


Figure 5.1: A factored graph representation of a title of a citation (PubMed Identifier [PMID] = 4429579). Each vertex has two factors: a lexical factor (concept name) and a semantic type factor. Abbreviations: qlco (Qualitative Concept), sbst (Substance), orch (Organic Chemical), moft (Molecular Function).

Graphs were categorized according to five major organism groups (invertebrates, birds, fish, fungi and humans), which were based on categorizations from the MeSH hierarchy. Within each group, graphs were labeled with a disease class label (e.g. Neoplasms, Bacterial infections and Mycoses, and Immune System diseases). After generation of concept graph representations for sentences, the next step was to apply a graph pattern mining method to find significant patterns. Fig. 5.1 shows an example of a concept graph generated for a sentence.

5.3.3 Graph Pattern Mining

A graph pattern analysis method was used to analyze concept graphs and extract subgraph pattern features that correlated with diseases of interest. A statistical model measured the quality of subgraph features in the graph datasets and a heuristic search algorithm used

this model to explore and evaluate the space of subgraph patterns within the graph dataset. This model was previously developed to analyze a graph dataset of human disease genetic pathways (Nabhan and Sarkar 2012), and extended in this study to allow for inclusion of multiple features per vertex in the concept graph datasets. The model used in this study is termed the Factored Graph Pattern Analysis Model (FGPAM). The model assumed a factored representation of vertices, with each vertex having a set of factors (e.g. label, part-of-speech-tag, and UMLS semantic type). One of the parameters of the method is a vector of weight values, one per factor, which represents the importance of the factor. For instance, more weight can be assigned to the semantic type factor and less weight to the part-of-speech factor. The parameters of the statistical model were estimated using the Expectation Maximization (EM) algorithm.

Graph Partitioning

The notion of graph partitioning denotes a function that maps edges of a graph into a range of subgraphs that can represent a candidate feature. These subgraphs are edge disjointed, but can share vertices. The set of subgraphs that are defined by the function π on a graph G is $H_\pi = \{g_i | \forall e \in E(g_i), \pi(e) = i\}$. There is a large space of possible partitionings and there was a need to measure the quality of each partitioning to search for best partitionings that highlighted key subgraph features in the graph datasets.

Each concept graph G was assigned a disease class label C. A set of possible partitionings of G was sought and a probability value was used to measure the quality of each partitioning. Partitioning quality was represented as a conditional probability ($P(\pi, G|C)$) of a partitioning π of a concept graph instance G given a class value C (Nabhan and Sarkar

2012, Nabhan and Sarkar 2013). The value $P(\pi, G|C)$ can be defined using the set of subgraphs $g \in H_\pi$as:

$$P(\pi, G|C) = P(g_1, g_2, ..., g_n|C) \tag{5.1}$$

It was assumed that subgraphs given a partitioning function on a concept graph were conditionally independent. Thus, it was possible to write $P(\pi, G|C)$ as:

$$P(\pi, G|C) = \prod_{g \in H_\pi} P(g|C) \tag{5.2}$$

The probability value $P(g|C)$ represented the degree to which a subgraph g was a feature of a class C. Subgraphs were approximated by a set of maximal paths connecting its vertices, and then the probability of a subgraph was computed as a function of the set of maximal paths. A maximal path was defined as a path that could not be extended by adding vertices to either end. The probability $P(g|C)$ was then expressed in terms of probabilities of maximal paths given a class C:

$$P(g|C) = P(a_1, a_2, ..., a_k \in g|C) \tag{5.3}$$

To simplify the computation of Equation 5.3, maximal paths were assumed to be conditionally independent:

$$P(g|C) = \prod_{a \in g} P(a|C) \tag{5.4}$$

The probability of a maximal path was computed as a function of annotations features for every vertex that lied in the path. Thus, a maximal path was factored into a set of

annotated sequences, each for an annotation factor. For instance, if there was a lexical factor representing a word token and a semantic type factor of a UMLS concept, then a sequence of tokens and a sequence of UMLS concepts were generated. Then, the probability $P(a|C)$ was represented as

$$P(a|C) = w_1 \times P(f_1|C) + ... + w_m \times P(f_m|C) \tag{5.5}$$

where $f_1 ... f_m$ represented annotated factored sequences for vertices that lie on maximal path a and $w_1 w_m$ are weights assigned to factors.

Using Equations 5.2 and 5.4, the likelihood of a partitioning π and a graph G instance given a disease class label C was written as

$$P(\pi, G|C) = \prod_{g \in H_\pi} \prod_{a \in g} P(a|C) \tag{5.6}$$

Equations 5.1-5.6 defined one way to measure the quality of key subgraph pattern features in a coherent way that took into account neighborhood of subgraph (and hence quality of a subgraph pattern was not computed in isolation) and also considered dependency on class labels assigned to graphs. A heuristic search function used Equations 5.1-5.6 to explore the space of possible partitionings and to find good subgraph patterns. This function was integrated into the EM algorithm during the process of estimating the parameters of the statistical model.

Parameter Estimation

The statistical model had a set of parameters θ:

$$\theta = \cup_{j=1}^{m} \{P(f_j|C)\} \tag{5.7}$$

The parameters θ consisted of a set of conditional probability tables of factored annotation sequences. The parameter estimation procedure required counting the incidence of a factored annotated sequence in subgraphs of generated partitionings of a graph. Annotated sequences were weighted by the probability score of the graph partitioning within which that sequence was found. At the same time, the partitioning probability needed the conditional probability tables of the annotated sequences, which did not exist a priori. To solve this problem, an iterative procedure was used to estimate model parameters while searching for better partitionings. This iterative procedure had two steps. In the first step, the most recent values of model parameters were used in the search for a better partitioning. In the second step, weighted counts of annotated sequences were collected and normalized to produce new conditional probability tables.

Model parameters were estimated according to Equations 5.2 and 5.4 using the EM algorithm (Dempster et al. 1977). Initially, a set of random partitionings was generated and the maximal paths within these partitionings were collected and an initial probability distribution for $P(f_j|C)$ was created, where f_j denoted an annotated sequence of factor j. The parameter estimation process for this study had two basic steps. The first step was to search for better partitionings using the most updated version of the probability table $P(f_j|C)$ obtained in the previous iteration of EM. Factored maximal path parameters counts are collected from within graph partitionings. The counts of a parameter in one graph was thus calculated as follows:

$$c(f_j|C;G) = \sum_{\pi} P(\pi|G,C)N(f_j,G) \sum_{j} \delta(f,f_j)\delta(C,C_j) \tag{5.8}$$

Here, $N(f_j, G)$ is the number of incidences a factored maximal path f_j appeared in G (in different subgraphs of G), and δ is the Kronecker's delta function. Then, these counts were normalized to obtain an updated conditional probability table. The probability value $P(\pi|G,C)$ is the normalized partitioning probability, and it was obtained by dividing $P(\pi, G|C)$ by the marginalized sum of probabilities other partitionings of G. In the second step, counts of factored maximal path parameters $c(f_j|C;G)$ were normalized to get $P(f_j|C)$.

Searching for Best Partitionings

At each iteration, the hill-climbing algorithm was used to search for better partitionings of the EM algorithm. Partitionings were presented as integer arrays with indexes referring to edges and entries referring to a subgraph that contains the edge indicated by the array index. This representation allowed for generation of new partitionings that could be evaluated using the most recent values of model parameters θ. Changing the entry value of the integer array reflected the shrinking of a subgraph, the growth of another graph, and could mean the split of a subgraph graph into two smaller subgraphs. For every pair of edges, a connectivity test was performed to determine if these edges shared a vertex. If two edges shared a vertex, then one of the edges propagated its subgraph id to the other.

There were two outcomes of the EM parameter estimation algorithm: (1) a set of conditional probability tables for annotation factors (model parameters); and, (2) a set of best partitionings of each concept graph in the dataset. A java-based tool was developed

to estimate FGPAM model parameters and find the best set of partitionings (highlighting significant patterns) for a given graph dataset. In addition to model parameters, a set of application parameters enables configuration of the tool according to the graph datasets (for instance, to determine whether edges are directed or undirected and to determine number of EM iterations). After FGPAM parameter estimation, the set of best partitionings were processed to extract significant subgraph patterns.

There might be thousands of partitionings for a given graph; each of which has been assigned a probability score determined by Equation 5.6. In previous work (Nabhan and Sarkar 2013), a defined threshold value was used to select high probable partitionings for subgraph pattern extraction. One potential problem with using probability threshold to select partitionings was that there was a wide-range variability of partitionings probability according to graph size (partitionings of large graphs tend to have lower probability value than those of smaller graphs). To solve this problem, partitionings were grouped by graph size (number of edges). Then, within each group (bin), partitionings were ranked from highest probability to lowest probability. Then, a threshold value t on rank can be set to select top-t partitionings. These high-rank partitionings were then used to extract key subgraph patterns. These patterns summarized information content regarding biological processes related to diseases.

5.3.4 Assessment of Model Organisms

The assessment of potential model organisms was performed using key subgraph patterns that were highlighted in the best partitionings of annotation-rich concept graphs of text sentences. For each disease category, subgraph patterns in each graph dataset of

a non-human organism group were matched to the corresponding disease patterns in the human dataset. An approximate graph matching method was used to measure the similarity between two subgraph patterns. The group organism with the highest subgraph matching score was deemed the best-fit model organism for the given disease category.

The approximate subgraph matching method used information of vertex connectivity and its semantic type factor. Given two subgraphs, vertices in the first subgraph were mapped to vertices in the second subgraph so that link information was preserved. Then, semantic annotations of vertices were matched. If the percentage of matching semantic annotations was above a specified threshold parameter, the two subgraphs were reported as similar.

5.4 Results

Concept graph representations of selected MEDLINE citations were constructed using a set of transformation rules from MMO output, resulting in approximately nine million concept graphs. The citations were selected based on organism groups, disease groups, and Biological Phenomena and Processes MeSH descriptors. The FGPAM java tool was then used to analyze the graph dataset to highlight significant patterns in concept graphs. These patterns were used as a basis to compare organisms for suitability as potential disease models.

5.4.1 Datasets and FGPAM Software Tool Parameters

Approximately nine million sentences were processed to generate the concept graphs. Sentences in this corpus were categorized into five organism groups as well as six disease categories. Table 5.2 shows the distribution of concept graphs across organism and disease groups.

Table 5.2: Number of concept graphs of each MeSH organism group distributed over six MeSH disease groups.

	Cardiovascular	Immune System	Nervous System	Viral	Endocrine System	Bacterial
Humans	3.55M	2.68M	3.1M	536K	1.46M	1.16M
Birds	5K	10K	11K	31K	2.6K	26K
Fishes	2K	1.3K	5K	7.7K	2K	12K
Fungi	1K	5K	2K	1.3K	0. 9K	26K
Invertebrates	1.5K	8.3K	11K	11K	6K	15K

Parameter estimation of FGPAM was performed on each dataset (specified by an organism group and a disease group). A set of application parameters of the FGPAM java tool needed to be determined to achieve the best results. One of the parameters was the vector of weight values of factored representation. To get better generalization capability of the pattern, more weight was given to the semantic type factor. This gave the semantic type factor an advantage when evaluating partitionings. Thus, subgraph patterns with identical structures (link) and semantic type factors would look more similar, even if the lexical factors (words) were different. The software implementation of FGPAM has a set of application parameters that needs to be adjusted for training data. For instance, there

are parameters to specify the maximum subgraph pattern size, the number of iterations of the EM algorithm, and weights of factors. The set of application parameters is given in Appendix B, Table B.1.

Given that the size of graph datasets can range from thousands to millions of concept graph items, it was necessary to divide the data into tractable item sets of up to 10,000 items. The processing steps estimation of FGPAM parameters, ranking of graph partitionings, and extraction of significant patterns from within partitionings were then applied to the smaller item sets. It took an average running time of two hours to finish the EM parameter estimation algorithm on a item sets containing 10,000 items. To facilitate the speed of data processing, item set processing was distributed across multiple processors at the Vermont Advanced Computing Core (VACC) facility. At the end of EM parameter estimation procedure, subgraph patterns were extracted from within the set of best partitionings of graphs. Numbers of extracted subgraph patterns (disease fingerprints) per organism group across six disease groups are shown in Table 5.3.

Table 5.3: Number of extracted subgraph patterns (disease fingerprints) per organism group across six disease groups analyzed for this study.

	Cardiovascular	Immune System	Nervous System	Viral	Endocrine System	Bacterial
Humans	0.14M	0.11M	0.14M	0.23M	0.05M	0.04M
Birds	2.5K	5.5K	5K	4.7K	1.2K	4K
Fish	0.7K	0. 5K	2.4K	3.5K	0.7K	4.8K
Fungi	0.3K	2K	0.6K	0.3K	0.2K	0.2K
Invertebrates	0.5K	4K	6K	5K	2.5K	7K

Table 5.4: Proportions of human-matched subgraph patterns (disease fingerprints) per organism group across six disease groups analyzed for this study.

	Cardiovascular	Immune System	Nervous System	Viral	Endocrine System	Bacterial
Birds	0.0003	0.0027	0.0225	0.0	0.0014	0.0012
Fishes	0.0094	0.0140	0.0003	0.0005	0.0050	0.0011
Fungi	0.0008	0.0001	0.0	0.0	0.0006	0.0
Invertebrates	0.0070	0.0003	0.0001	0.0002	0.0	0.0007

5.4.2 Emergence of Patterns of Biological Phenomena

Subgraph patterns of each candidate organism group were matched to subgraph patterns of the human dataset for each of the disease groups examined by this study. Two subgraph patterns were considered a match if there was a one-to-one mapping between the vertices of the two graphs such that vertex degrees and edges were completely matched, and additionally, the proportion of matching between factored annotations of semantic type of corresponding vertices was above 0.65. This matched procedure was repeated for each organism group and disease group.

The factored graph representation that included semantic types improved pattern generalization, as demonstrated by the matched subgraph patterns that did not agree much on the lexical factor (first factor); however, the factored graph representation largely agreed on the semantic factor (second factor). For instance, Figure 5.2 shows an example of matched subgraph patterns for cardiovascular diseases. While there were slight differences

between lexical factors of vertices, there were similarities between semantic factors.

5.4.3 Assessment of Model Organisms

For this study, a total of 82 potential model organisms were evaluated based on a similarity score between pairs of subgraphs, computed as the proportion of vertex factors with the same semantic type for two subgraphs. Only pairs of subgraph patterns with similarity score above the specified threshold parameters were considered as matched. Assessment of potential model organisms was based on the proportion of subgraph patterns in human datasets with match to the given organism groups subgraph patterns. Table 5.4 shows performance of each organism group relative to each disease group analyzed. Tables 5.A. and 5.B. show detailed scores of the top three model organisms covered in the datasets regarding cardiovascular and immune system diseases. Appendix A contains tables with extended results for these two diseases as well as for nervous system, endocrine system, bacterial and viral diseases.

For cardiovascular diseases, the trout fish performed the best as a potential model organism (Table 5.5.); the torpedo fish had the best number of matches for immune system diseases (Table 5.6.). For nervous system diseases, chicken had the best number of matches (Appendix Table B.4.). Overall, a fewer number of organism fingerprints matched the human fingerprints for viral diseases (Appendix Table B.4.) and bacterial diseases (Appendix Table B.6.). Finally, zebrafish performed the best as a potential model for endocrine system diseases (Appendix Table B.7.).

Table 5.5: Detailed fingerprint matching scores of model organisms for Cardiovascular Diseases.

Cardiovascular Diseases	
MeSH Organism Group MeSH Organism	Number of matched fingerprints
Fishes	
Trout	861
Zebrafish	366
Salmon	74
Invertebrates	
Diptera	634
Urochordata	167
Ticks	82
Birds	
Parrot	11
Columbidae	10
Chicken	9
Fungi	
Cryptococcus	92
Candida albicans	14
Polyporales	14

5.5 Discussion

A graph-based method of text pattern mining was developed for assessment of organisms as disease models based on evidence found in biomedical literature. A first step in the method was to generate a graph-based representation for text sentences using a set of transformation rules with patterns that matched the syntactic structure of the sentences. Rich syntactic and semantic annotation sets for biomedical citations were extracted from available MetaMap machine output files and used to generate graph representations for citations. A knowledge-rich annotation scheme was developed to assign vertices to

Table 5.6: Detailed fingerprint matching scores of model organisms for Immune System Diseases.

Immune System Diseases	
MeSH Organism Group MeSH Organism	Number of matched fingerprints
Fishes	
Torpedo	1481
Zebrafish	20
Carps	18
Invertebrates	
Nippostrongylus	13
Cockroaches	6
Anisakis	5
Birds	
Chicken	284
Columbidae	4
Duck	3
Fungi	
Spores, Fungal	10

multiple annotation types, termed factors. These annotation-rich concept graphs provided an opportunity to apply graph pattern mining methods to information inferred from biomedical literature. A statistical pattern analysis model was developed to provide a quantitative measure of pattern quality. A heuristic search algorithm used this model to find key patterns in graphs while estimating the parameters of the statistical models. This method was applied to graph datasets with nine million items and was used to address the problem of assessment of potential model organisms using evidence in biomedical literature.

CHAPTER 5. GRAPH-BASED MINING IN BIOMEDICAL LITERATURE

Graph-based representation of biomedical texts allowed for using a graph pattern analysis method to capture complex relationships between concepts in text sentences. Subgraph patterns that are semantically similar (sharing similar semantic types assigned to concepts) were used in this study to compare organisms to find better disease models. Subgraph patterns in citations that describe biological phenomena and processes in humans were used as a reference set for comparisons among organisms. Thus, the methods developed in this study enabled a shift from lexical patterns (patterns of words) to ones based on semantics (patterns of semantic types derived from a knowledge base such as an ontology). This shift from lexical patterns to semantic patterns may help with dealing with data sparsity issues, particularly as seen in textual data.

Knowledge-rich graph-based methods for analysis of patterns in text articles provide tremendous opportunities for analysis of the content of biomedical literature. Using multiple annotation types (factors) for graph vertices enabled the incorporation of relevant domain knowledge. The statistical model presented here allowed for the utilization of factored representation of graph vertices and the assignments of weights to factors based on importance/relevance of the factor to the task. While only two factors were used in the study (lexical and semantic annotation types), it might be useful to incorporate other potentially relevant factors (e.g. part-of-speech tags) into the graph representation for future studies. Subgraph patterns identified within graph representations of text sentences can be useful for a number of purposes. For instance, these patterns may be used for exploratory tasks, text categorization, and summarization of text content. In this study, subgraph patterns were used for evaluation of potential model organisms.

The organism that shared the highest number of patterns with a set of human disease patterns was reported as the best possible model organism. The results from this study suggest that traditional model organisms may not necessarily be the best models for some disease categories. For instance, the results show that trout was predicted to be a better model than more classical model organisms (e.g., zebrafish and fruit fly). Similarly, for immune system diseases, the torpedo fish was predicted to be better model than zebrafish. These results demonstrate the potential of using graph-based text mining techniques to assess organisms as disease models.

Recent studies have addressed the question of finding better model organisms for human diseases that expand beyond the classically used models (especially mouse and rat). The notion of phenologs (phenotypes that are equivalent across organisms) has been used to search for better model organisms. For instance, McGary el al. used a phenology approach to suggest a worm model for breast cancer, a yeast model for angiogenesis disorders, and a plant model for Waardenburg syndrome (McGary et al. 2010). In another recent study, graph pattern mining of biological interaction networks demonstrated the ability to evaluate candidate organisms that do not always suggest a classic model organism (Nabhan and Sarkar 2013). Finally, Karathia et al. have developed a strategy for evaluation of organisms using functional classification of proteins and proteome analysis (Karathia et al. 2011).

Knowledge-rich models can address the data sparsity problem, especially in text data. Using a multi-factor annotation scheme for graph vertices provided an opportunity to manage knowledge content of text in a more structured and visual-friendly way. A graph-based

statistical model enabled the adjustment of vertex annotation factor weights according to the problem domain of interest. This knowledge-rich model therefore made it possible for two lexically different, but semantically equivalent subgraph patterns to be matched. Semantically equivalent patterns formed the basis at which organisms were assessed for being potential disease models.

5.6 Conclusion

In addition to direct biological information, such as molecular sequence, phenotype information, pathways, biomedical literature resources provide a rich source of additional information, including molecular, phenotypic and procedural information that can be correlated to disease. Genomic and phenotypic materials have long been the focus when studying a biological process in an organism. This study demonstrated the possibility to aggregate information about biological processes and phenomena embedded in biomedical literature and use it for assessment of organisms as disease models. To appreciate the full range of possible model organisms that may be suitable for the study of a particular disease, it may be of value to integrate literature based inferences and biological data for evaluating model organism suitability.

5.7 References

Altman, R. B. and T. E. Klein (2002). Challenges for biomedical informatics and pharmacogenomics. *Annual review of pharmacology and toxicology 42*(1), 113–133.

Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research 32*(suppl 1), D267–D270.

Browne, A. C., G. Divita, A. R. Aronson, and A. T. McCray (2003). Umls language and vocabulary tools: Amia 2003 open source expo. In *AMIA Annual Symposium Proceedings*, Volume 2003, pp. 798. American Medical Informatics Association.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–38.

Glass, C. K., K. Saijo, B. Winner, M. C. Marchetto, and F. H. Gage (2010). Mechanisms underlying inflammation in neurodegeneration. *Cell 140*(6), 918–934.

Karathia, H., E. Vilaprinyo, A. Sorribas, and R. Alves (2011). Saccharomyces cerevisiae as a model organism: a comparative study. *PloS one 6*(2), e16015.

Leskovec, J., M. Grobelnik, and N. Milic-Frayling (2004). Learning sub-structures of document semantic graphs for document summarization.

Lipscomb, C. E. (2000). Medical subject headings (mesh). *Bulletin of the Medical Library Association 88*(3), 265.

McGary, K. L., T. J. Park, J. O. Woods, H. J. Cha, J. B. Wallingford, and E. M. Marcotte (2010). Systematic discovery of nonobvious human disease models through orthologous phenotypes. *Proceedings of the National Academy of Sciences 107*(14), 6544–6549.

Nabhan, A. R. and I. N. Sarkar (2012). Mining disease fingerprints from within genetic pathways. In *AMIA Annual Symposium Proceedings*, Volume 2012, pp. 1320. American Medical Informatics Association.

Nabhan, A. R. and I. N. Sarkar (2013). Structural network analysis of biological networks for assessment of potential disease model organisms. *Journal of Biomedical Informatics*.

Rebholz-Schuhmann, D., A. Oellrich, and R. Hoehndorf (2012). Text-mining solutions for biomedical research: enabling integrative biology. *Nature Reviews Genetics*.

Sandelin, A., W. W. Wasserman, and B. Lenhard (2004). Consite: web-based prediction of regulatory elements using cross-species comparison. *Nucleic acids research 32*(suppl 2), W249–W252.

Whelan, R. S., V. Kaplinskiy, and R. N. Kitsis (2010). Cell death in the pathogenesis of heart disease: mechanisms and significance. *Annual review of physiology 72*, 19–44.

Wolf, Y. I., I. B. Rogozin, A. S. Kondrashov, and E. V. Koonin (2001). Genome alignment, evolution of prokaryotic genome organization, and prediction of gene function using genomic context. *Genome Research 11*(3), 356–372.

Yoo, I., X. Hu, and I.-Y. Song (2007). A coherent graph-based semantic clustering and summarization approach for biomedical literature and a new summarization evaluation method. *BMC bioinformatics 8*(Suppl 9), S4.

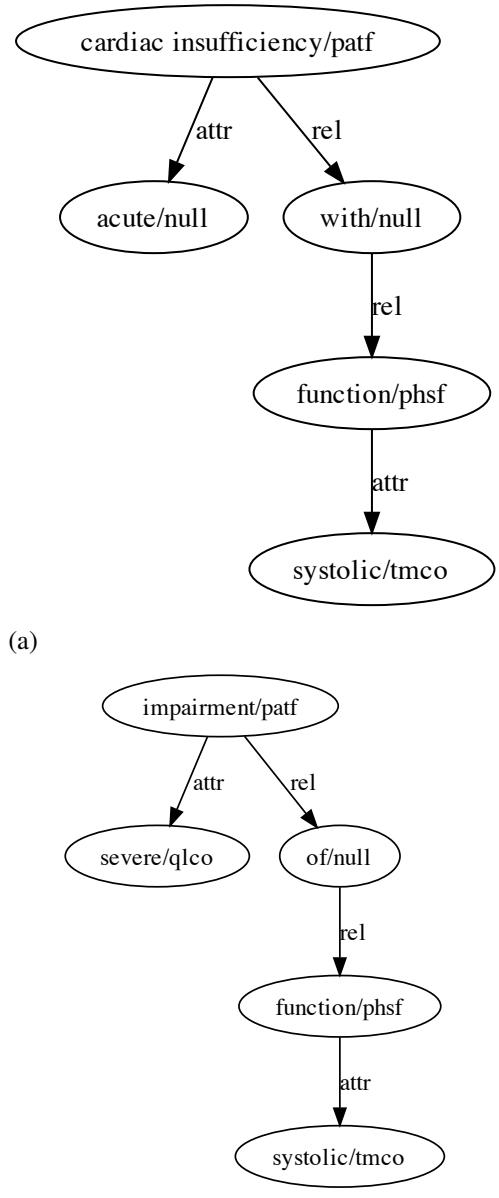


Figure 5.2: Two matched subgraph patterns of Cardiovascular Diseases. (a) a human subgraph pattern (PubMed Identifier [PMID]=14571638). (b) a *Drosophila melanogaster* subgraph pattern (PubMed Identifier [PMID] = 16432241).

Chapter 6

Concluding Remarks

In this dissertation, four studies on methods for data mining of biological networks, chemical compounds and biomedical literature were presented. The promising results of this research collectively have enhanced the state-of-the-art for data mining methods and advanced our understanding for their potential in biomedical research.

6.1 Summary and Conclusions

In Chapters 2 and 3, the research was focused on the development of a statistical graph pattern analysis model that enabled simultaneous and coherent searches for key subgraph patterns in graphs based on the notion of graph partitioning. The probability value of observing a partitioning was computed as a function of the probability values of its constituent subgraphs, emphasizing the importance of pattern context in the extraction of key patterns from within graphs. A key property of the developed model is the use of maximal paths to approximate subgraph patterns. This property of the model allowed for

effective analysis of small datasets of sparse graphs. The Expectation Maximization (EM) algorithm was used to estimate the model parameters. A heuristic search algorithm was developed to use this model to explore a large space of possible partitionings of graph items using a pattern growth technique.

It was shown, based on the overall promising performance in graph classification tasks, that the developed method of simultaneous, context-aware search for patterns can yield a performance that is: (1) superior to frequent pattern mining methods; and, (2) competitive with graph kernel methods. The method demonstrated flexibility when tested on different genres of data (directed and undirected graphs as well as balanced and imbalanced datasets). The results of the performance evaluation suggest that simultaneous and coherent search for patterns are imperative when analyzing sparse, limited size datasets of complex structures such as graphs.

The developed method was applied to the problem of analyzing genetic pathways of human diseases to identify potentially significant patterns (disease fingerprints), which is a significant research problem in biomedical research. Genetic pathway graph datasets generally have a limited number of items with a diverse set of vertex labels (i.e., the alphabet of vertex labels may consist of thousands of gene names). Thus, the use of maximal paths to approximate subgraphs were shown to be useful when analyzing sparse and limited-size genetic pathways, since paths tended to be more frequent than subgraphs in that case. This graph-based method may provide an effective means to analyze genetic pathways in a way that enables a shift from single gene-based analysis to a system-level analysis of disease genes. This method may thus be of particular relevance in the analysis

of complex diseases or traits.

In Chapter 4, the developed pattern analysis method was applied to the biomedical research problem of searching for the best organism to be used as a potential disease model. Genetic pathways, which describe biological processes and gene interactions related to human diseases, were analyzed with the aim to identify functional and structural patterns (disease fingerprints). A knowledge-based annotation scheme was applied to pathway graph data for annotating genes/proteins vertices with molecular function annotations that were imported from the Gene Ontology (GO) annotation knowledge base. The resulting subgraph patterns were thus patterns of molecular functions (i.e., not gene/protein names). These abstract GO-annotated patterns therefore enabled the summarization of the molecular ingredients of biological processes related to diseases.

The identified disease fingerprints were used to predict genetic pathways in large biological interaction networks for a number of organisms. Nodes of each interaction network were annotated with molecular function annotations from GO, and therefore enabled the matching of genetic pathway disease fingerprints to subnetworks within organism interaction networks. A graph indexing and query processing method was developed to allow for efficient search in the interaction networks. The accuracy of predicting new pathways within interaction networks was assessed using a set of reference (published) pathways for each organism analyzed in the study. The pathway prediction accuracy measurement presented an objective assessment of organisms as potential disease models based on the human disease fingerprints. The results suggest that using knowledge-rich graph-based models for searching for model organisms may be an effective means that

may complement traditional orthology-based methods.

Finally, an annotation-rich graph-based method for the analysis of text patterns in biomedical literature was developed and evaluated in Chapter 5. Sentences were mapped into graphs using text annotations obtained from an existing Natural Language Processing (NLP) tool (MetaMap). The morpho-syntactic and semantic information generated for sentences enabled a feature-rich annotation set for graph vertices. These vertex annotation types (factors) allowed for the incorporation of multiple knowledge sources to get better pattern quality. The graph pattern analysis model was enhanced to allow for a factored-based analysis of subgraph patterns. This combined NLP-graph-based method had the advantage of handling data sparsity issues (particularly of text data) and enabled identification of semantically similar, lexically different subgraph patterns. This set of semantic patterns summarized information content in text.

The problem of assessing potential disease model organisms was revisited through the application of the graph-based text analysis method to biomedical literature data. The organism assessment was based on evidence collected from biomedical abstracts (as indexed and available from MEDLINE). Semantic patterns were identified in biomedical abstracts that focused on disease biological processes and phenomena in humans as well as 82 non-human organisms. These organisms were evaluated as disease models based on similarity of their semantic patterns to semantic patterns identified in biomedical abstracts about human diseases. Based on the results, graph-based methods for the analysis of textual data may be a promising knowledge-rich approach for corpus linguistics and text mining tasks.

These methods might be integrated with comparative genomics methods to possibly enable better understanding of disease-related biological processes.

6.2 Future Work

The dissertation was primarily focused on the investigation of methods to enable an important problem in the biomedical research: the assessment of disease model organisms. Graph-based analysis methods were applied to analyze diverse genres of data of varying complexities (biological networks, chemical compounds, and textual data). Graphs are powerful data models that can to represent complex relationships in for a given problem domain. Semantics inherent in graph data can be captured based on the notion of vertex connectivity. The analysis of graph patterns faces major challenges including : the high dimensional feature space and data sparsity. These challenges open future directions for some interesting future work.

First, there is a growing need to develop graph analysis methods to process large amounts of graph items available in (possibly unbounded) graph data streams, in which graph edges are received and updated in a sequential manner in the form of a stream. These graph streams are abundant in dynamic applications of social networks and the World Wide Web. A key characteristic of a graph stream is its continuous update and the high speed of incoming edges. This poses a computational challenge to graph pattern analysis methods. A n interesting future research problem is may be the design of efficient methods that can incorporate knowledge bases for data mining over graph streams.

Second, while the graph pattern analysis models in this dissertation were developed to analyze graph datasets (of thousands of graph items), a more general approach is to modify these models to for analyzing data that are is present in the form of a single large graph item (having tens or hundreds of thousands of vertices). It would be interesting to redefine graph partitioning in that case. Given the prohibitively large number of potential subgraph patterns in a large single graph item, the strategy of simultaneous search for patterns needs to be modified significantly. To this end, graph clustering and graph modularity methods may can be used as a pre-processing step to find a startup, initial graph partitioning at low computational cost. Then, the method can be modified in that case so that the basic unit in the models would be graphlets or motifs (small subgraphs), in contrast to using edges as basic units.

Third, the breadth of the developed methods in this dissertation can may be extended through the application of these methods to study new problems in other domains. These The developed methods could can be applied to: (1) finding meaningful patterns in textual content of social networks (e.g., Facebook and Twitter), (2) to identifying online search patterns in users search logs, (3) to identifying patterns of user preferences for use in recommendation systems, and (4) to identifying malicious software patterns in system call diagrams in computers operating systems. Finding appropriate domain knowledge sources to annotate graph vertices can may be a critical issue in the success of these methods to highlight meaningful patterns in data.

Appendix A: Model and Algorithm Details

A.1 Preliminaries and Notations

A graph G consists of a set of vertices V and a set of edges E in which each edge $e \in E$, denoted by $e(u, v)$, links two vertices $u, v \in V$. A subgraph consists of a set of nodes $V \subseteq V$ together with a set of edges $E \subseteq E$ that links its nodes. In this study, genetic pathways were modeled as a set of labeled directed graphs.

Definition A.1 *Labeled Graph. A labeled graph $G(V, E, L_V, L_E, \sum_V, \sum_E)$ has a node labeling function $L_V : V \rightarrow \sum_V$ that assigns labels from a node alphabet set $\sum_V$ to nodes and an edge labeling function $L_E : E \rightarrow \sum_E$ that assigns labels from an edge alphabet set $\sum_E$ to edges. A labeled subgraph g consists of a subset of nodes of G and edges that link them. Labels of nodes and edges of a subgraph g are the same as its super graph G.*

The node alphabet set $\sum_V$ contained GO terms. The edge alphabet set $\sum_E$ contained relation types in KEGG disease pathway dataset. The basic definition of a labeled graph was extended in two ways. First, a NULL label ϵ was assigned to gene nodes with no GO

terms associated. Second, an entity could be mapped to more than one label. Also, edges in a given pathway could be labeled with more than one relation type.

Disease fingerprints are subgraphs of GO annotated disease pathways graphs that were are assumed to represent functional sub-processes that could be characteristics of a disease class such as immune, infectious, or neurodegenerative disease. Disease fingerprints are therefore functional structural patterns in GO annotated graphs. To quantify the degree to which a fingerprint was related to each disease class, a first step was to use a utility function to highlight a set of subgraphs (fingerprints) in a given pathway graph. This utility function was termed a partitioning function.

Definition A.2 *Partitioning function π. Let $E(.)$ denote edge set of a graph G. A partitioning function $\pi : E(G) \rightarrow Z$ assigns an integer to every edge of G such that edges with the same integer form a subgraph. The set of subgraphs H highlighted by a specific partitioning function π is defined as $H_\pi = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \pi(e) = i\}$.*

Figure 4.2 illustrates the concept of partitioning. From the above definition, it follows that every edge must be covered by only one subgraph (i.e., subgraphs of a given partitioning are edge-disjoint). There is a big space of partitioning functions for each graph in the dataset and this space is not known a priori. Searching for good partitioning functions was thus one of the objectives of this study. Preventing subgraph overlapping has a useful impact on speed and memory during search for partitioning for each graph. For instance, the probability estimation algorithm (presented in the next section) does not have to minimize overlapping of subgraphs while searching for partitionings. To accommodate side effects of this restriction, the process of identifying disease fingerprints takes into account information from many hypothesized partitioning functions for every graph in the dataset. In this study, partitionings were represented by integer arrays where indices represent edge identi-

fiers and values represent subgraphs to which an edge belongs. This compact representation allowed for easy extension of partitionings by modifying edge-to-subgraph assignments of an existing partitioning in order to generate new ones. This array representation was also helpful in detecting similarity between partitionings (which was useful in minimizing memory requirements of the tool by keeping only one copy of a partitioning among several equi-probable partitionings).

A.2 Mathematical Model

Graphs in the design dataset were assumed to be independent and identically distributed (iid) data observed from an unknown probability distribution $P(G)$. The iid data assumption was made for the purpose of facilitating statistical inference and to make decision about properties (e.g., class label) of a graph instance independent of other graph instances in the dataset. For each pair of graph G and disease class C, a probability value was used to quantify the relation between a graph and its class label. Let the probability value $P(G|C)$ quantify the characteristics of class C that is observed in graph G. Modeling this probability value directly can be hard, mainly because: (1) it is a computationally non-trivial task to determine if two graph instances are equal using the graph isomorphism test (Read and Corneil 1977, Shang et al. 2008); and (2) due to the data sparseness problem (it is usually hard to find more than one isomorphic instance of the same graph in a given dataset). An indirect way to model $P(G|C)$ was used to provide the model with access to GO functional annotations as well as hidden structural patterns (collectively referred to as fingerprints) in a given graph. Using the utility of partitioning function, a more useful probability value $P(G, \pi|C)$ would involve a graph instance G, a class label C, and a graph partitioning that divides G into a set of fingerprints. $P(G, \pi|C)$ quantifies the probability of observing struc-

tural patterns of class C in graph instance G. There are many possible partitionings for the same graph instance, and to take into account all possible structural patterns represented in these partitionings, the probability value $P(G|C)$ can be expressed as

$$P(G|C) = \sum_{\pi} P(G, \pi|C) \tag{A.1}$$

Let H_π be the set of subgraphs according to a partitioning function π of graph G: $H_\pi = \{g_i \mid \forall e \in E(g_i), E(g_i) \subseteq E(G), \pi(e) = i\}$

Assuming that subgraphs resulting from a partitioning function are conditionally independent, $P(G, \pi|C)$ can be written as

$$P(G, \pi|C) = \prod_{g \in H_\pi} P(g|C) \tag{A.2}$$

The probability value $P(g|C)$ represents the likelihood that subgraph g is a characteristic structural pattern of class C. Here, it should be pointed out that the conditional independence assumption made here is mathematically plausible considering that: (1) subgraphs in one partitioning do not overlap (i.e., do not share common edges, according to definition of π); and (2) this assumption is made for subgraphs within the same partitioning (i.e., it is local to a specific partitioning, not for all combinations of subgraphs.) For the purpose of probability estimation, counting the number of instances of a subgraph in all partitionings of graph dataset is impractical, since it re-introduces the problem of subgraph isomorphism (Read and Corneil 1977). In this study, GO- annotated maximal paths inside the subgraphs were used to approximate representation of subgraphs. Each maximal path represented a sequence of GO annotations of nodes that lay in that maximal path. In case a node has more than one GO annotation, multiple maximal paths are generated so that

each maximal path has only one GO annotation per node. Then, $P(g|C)$ was calculated approximately as

$$P(g|C) = \prod_{a \in g} P(a|C) \tag{A.3}$$

where a denotes a GO-annotated maximal path that connect a subset of nodes inside subgraph g. Using Equation A.3, the likelihood of a partitioning and a graph instance given a disease class label can be written as

$$P(G, \pi|C) = \prod_{g \in H_\pi} \prod_{a \in g} P(a|C) \tag{A.4}$$

and, finally, $P(G|C)$ can be expressed as

$$P(G|C) = \sum_{\pi} \prod_{g \in H_\pi} \prod_{a \in g} P(a|C) \tag{A.5}$$

Thus, Equations 2-5 casts the problem of searching for disease fingerprints as estimating a conditional distribution of GO annotated maximal paths given disease classes, while maintaining a set of best partitionings for each graph instance highlighting disease fingerprints.

A.3 Probability Estimation and Searching for Best Partitionings

For a given pathway design dataset, two data entities need to be generated: (1) the best scoring partitioning set (that contains disease fingerprints within each pathway); and (2) the

conditional probability table $P(a|C)$. The generation of each of these two entities requires the existence of the other, but neither of them exists with the graph data at the beginning of probability estimation process. Therefore, both entities must be generated initially at the same time, albeit with low likelihood, and probability estimate of $P(a|C)$ and partitionings likelihood values can be improved iteratively. Here, the estimation of model parameters θ is performed following a maximum likelihood approach using the Expectation-Maximization (EM) (Dempster et al. 1977, Moon 1996) algorithm. Model parameters consisted of the probability distribution of maximal paths given class labels:

$$\theta = \{P(a|C)\} \tag{A.6}$$

There can be a large space of possible values of the parameters θ and the search for best parameter values can be based on maximizing the likelihood on the graph dataset:

$$\theta^* = argmax_\theta\{\prod_{n=1}^{N}[P_\theta(G_n|C_n)]\} = argmax_\theta\{\prod_{n=1}^{N}[\sum_{\pi} P_\theta(G_n, \pi|C_n)]\} \tag{A.7}$$

where N is the number of graphs in the dataset and P?(Gn, —Cn) is computed by Equation A.4 and the probability distribution $P_\theta(a|C)$. $P_\theta(G, \pi|C)$ represents the probability of a partitioning of a graph given a class label using a given set of values of parameters θ. The EM algorithm aims at maximizing the likelihood function in Equation A.7 while identifying best graph partitionings that highlight key patterns. Because it was computationally expensive to consider all possible partitionings for graphs in the probability estimation algorithm, a priority queue of a limited number of highly probable partitionings was maintained. In each iteration, searching for new partitionings extends the set of best partitionings of each

graph. These partitionings are evaluated using Equation A.4 and the parameter values θ obtained in the previous iteration. An initial set of random partitionings is generated for each graph in the dataset. Annotated maximal paths were extracted from each subgraph of a given partitioning and the parameters θ are initialized with uniform probability values. The EM algorithm consisted of repeated iterations of E-Step and M- Step. In the E-Step of the algorithm, and for each graph, maximal path parameter counts are collected from within partitionings. The count of a parameter in one graph is calculated using:

$$c(a|C;G) = \sum_{\pi} P(\pi|G,C)N(a,G)\sum_{j}\delta(a,a_j)\delta(C,C_j) \tag{A.8}$$

Here, $N(a,G)$ is the number of times a maximal path a appeared in G (in different subgraphs of G), and δ is the Kronecker's delta function. The probability value $P(\pi|G,C)$ is the normalized partitioning probability conditioned on a graph and a class and is given by:

$$P(\pi|G,C) = \frac{P(G,\pi|C)}{\sum_{\pi'} P(G,\pi'|C)} \tag{A.9}$$

where $P(G,\pi'|C)$ is given by Equation A.4. The summation in Equation A.10 is over the set of best partitionings that is generated for graph G. Since this set is limited in size, Equation A.10 is only an approximation of partitioning quality. Multiplying the path-class counts $\delta(a,a_j)\delta(C,C_j)$ by partitioning probability $P(G,\pi|C)$ in Equation A.8 aimed at weighing each parameter count according to partitionings quality represented by $P(G,\pi|C)$. In the M-step, the maximal path parameters are computed by normalizing the counts:

$$P(a|C) = \frac{\sum_n c(a|C;G_n)}{\sum_{n,a} P(a|C;G_n)} \tag{A.10}$$

For each iteration of the model training algorithm, a search for best partitionings is performed using the best parameters θ estimated so far. Generating new partitionings from existing partitionings can be achieved moving edges from one subgraph to another subgraph. This way some subgraph patterns can grow while others can diminish. Figure A.1 illustrates the process of generating a new partitioning from an existing one. Both existing and newly generated partitionings were evaluated using Equation A.4 based on the most recent parameter values θ. In this study, the parameter estimation algorithm was run four iterations.

In summary, the model training procedure aimed to estimate the probability distribution $P(a|C)$. As a by-product of this procedure, a set of best partitionings of each graph highlighted the key subgraph patterns in the dataset. The pattern analysis model described above was used to find best partitionings in disease pathways with nodes annotated with molecular functions. Key patterns were extracted from best partitionings of pathways to be matched to sub-networks in gene/protein interaction network of a species.

A.4 An Algorithm for Matching Query Subgraphs to Interaction Networks

The following algorithm (shown in Table A.1) shows the three steps of matching a query subgraph (disease fingerprint) to an interaction network using a network index.

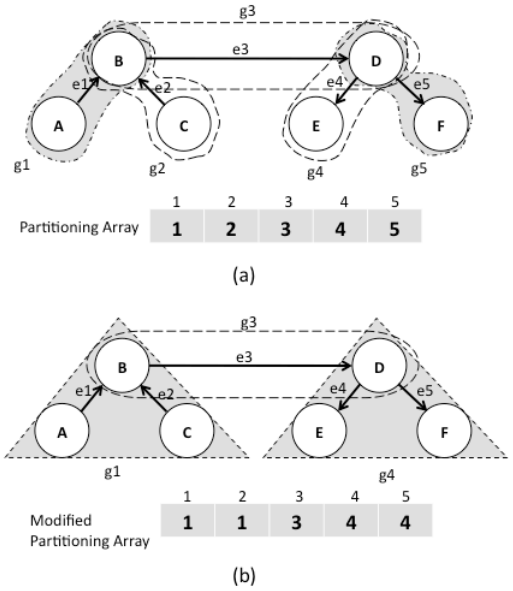


Figure A.1: A labeled directed graph that represents a functionally annotated genetic pathway.

Table A.1: An algorithm for matching query subgraphs to interaction networks

Input: Network index: index, network adjacency matrix; subgraph pattern: g, V(g) vertex set of g
Process
Step 1:Initialization
1: for each node $v \in V(g)$
2: initialCandidateMatchingSet(v) =
3: for each neighbor node u of v
4: $mSet =$
5: let $k \leftarrow (Lv(v), Lv(u))$
6: $vals$ = index.get(k)
7: for each $x \in vals$
8: $mSet$.insert(x)
9: initialCandidateMatchingSet(v).insert(mSet)
Step 2: Applying topological constraints
10: for each node $v \in V(g)$
11: let candidateMatchingSet(v) be intersection of all sets in initialCandidateMatchingSet(v)
12: for each node $v \in V(g)$
13: let S = candidateMatchingSet(v)
14: remove every item $i \in S$ if i is not linked to any item of candidate sets of neighbors of node v
15: return if S is empty
16: repeat 15-17 until no item can be removed from candidate sets
Step 3: Generate subnetworks by finding edges between nodes in final candidateMatchingSet
17: matchedSubNetworks = Array($|V(g)|$)
18: let S be the array of all nodes in $V(g)$
19: matchedSubNetworks ($S[1]$) =
20: for each network node identifier u in candidateMatchingSet of node $S[1]$
21: matchedSubNetworks ($S[1]$).append(u)
22: for $i = 2$ to $|S|$ // $|S|$ denotes size of set S
23: partialnetworks = matchednetworks($S[i-1]$)
24: for each partial network h in partialnetworks
25: for each network node identifier u in candidateMatchingSet of node $S[i]$
26: if $\exists$ node $w \in h$ such that u is linked to w in the interaction network
27: matchednetworks($S[i]$).append($u \cup h$)
Output: matchedSubNetworks[$|S|$] // output last element in the array matchedSubNetworks

Appendix B: Supplementary Materials on Results and Software Tool Parameters

B.1 FGPAM Java tool parameters

Table B.1: FGPAM Java Tool Parameters.

Parameter	Parameter Description	Value
$dataset\_name$	Graph data file name	filename_prefix
f	Number of factors	2
w	Weights of factors	$[0.3, 0.7]$
directed	Are edges directed?	True
$maxedges$	Max subgraph size	20
$maxpathlen$	Max number of vertices in a path	7
$prank$	Partitionings rank threshold	30
$minfeaturesize$	Min pattern size to report in results	4
n	Number of EM iterations	3

B.2 Detailed Result Tables of Model Organism Evaluation

Table B.2: Detailed fingerprints matching scores of organisms for cardiovascular diseases.

Cardiovascular Diseases	
MeSH Organism Group MeSH Organism	 Number of matched fingerprints
Fishes	
Oncorhynchus mykiss	861
Zebrafish	366
Salmon	74
Shark	18
Dogfish	5
Lamprey	2
Invertebrates	
Diptera	634
Urochordata	167
Ticks	82
Strongyloidea	60
Cockroaches	21
Acanthocephala	18
Bivalvia	6
Birds	
Parrot	11
Columbidae	10
Chicken	9
Coturnix	9
Turkey	5
Geese	1
Fungi	
Cryptococcus	92
Candida albicans	14
Polyporales	14

Table B.3: Detailed fingerprints matching scores of organisms for immune system diseases.

Immune System Diseases	
MeSH Organism Group MeSH Organism	Number of matched fingerprints
Fishes	
Torpedo	1481
Zebrafish	20
Carps	18
Electrophorus	9
Tilapia	7
Oryzias	5
Salmon	3
Invertebrates	
Nippostrongylus	13
Cockroaches	6
Anisakis	5
Spodoptera	4
Ascaris	2
Pyroglyphidae	1
Mites	1
Ceratopogonidae	1
Birds	
Chicken	284
Columbidae	4
Duck	3
Turkey	2
Quail	2
Parrot	2
Geese	1
Fungi	
Spores, Fungal	10
Blastomyces	1

Table B.4: Detailed fingerprints matching scores of organisms for nervous system diseases.

Nervous System Diseases	
MeSH Organism Group MeSH Organism	Number of matched fingerprints
Fishes	
Carps	15
Batrachoidiformes	10
Goldfish	9
Zebrafish	4
Invertebrates	
Culicidae	19
Diptera	1
Ixodidae	1
Birds	
Chicken	2407
Coturnix	320
Columbidae	135
Duck	104
Turkey	67
Geese	31
Sparrow	20
Parrots	17
Strigiformes	12
Finches	10
Parakeet	3
Spheniscidae	3
Fungi	
None	

Table B.5: Detailed fingerprints matching scores of organisms for viral diseases.

Viral Diseases	
MeSH Organism Group MeSH Organism	Number of matched fingerprints
Fishes	
Carps	60
Salmo salar	23
Cypriniformes	22
Oncorhynchus mykiss	20
Zebrafish	4
Salmonidae	2
Invertebrates	
Moths	27
Culicidae	13
Ticks	3
Ceratopogonidae	2
Phthiraptera	2
Birds	
Chickens	12
Ducks	1
Fungi	
Candida	3
Aspergillus fumigatus	1

Table B.6: Detailed fingerprints matching scores of organisms for bacterial diseases.

Bacterial Diseases	
MeSH Organism Group MeSH Organism	Number of matched fingerprints
Fishes	
Salmo salar	10
Oncorhynchus mykiss	6
Salmon	5
Cyprinidae	4
Flatfishes	3
Zebrafish	2
Tilapia	2
Gadiformes	2
Sea Bream	2
Goldfish	2
Ictaluridae	1
Invertebrates	
Ostreidae	8
Angiostrongylus	4
Ticks	2
Aedes	1
Anopheles gambiae	1
Siphonaptera	1
Birds	
Chickens	37
Turkeys	12
Ducks	1
Fungi	
None	

Table B.7: Detailed fingerprints matching scores of organisms for endocrine system diseases.

Endocrine System Diseases	
MeSH Organism Group MeSH Organism	Number of matched fingerprints
Fishes	
Zebrafish	177
Torpedo	19
Oncorhynchus mykiss	16
Salmon	13
Tilapia	12
Oryzias	8
Flounder	3
Catfishes	2
Invertebrates	
Butterflies	1
Birds	
Chickens	67
Coturnix	3
Fungi	
Basidiomycota	31
Candida albicans	3

Bibliography

Aigner, M. (1999). A characterization of the bell numbers. *Discrete mathematics 205*(1), 207–210.

Aitman, T. J., C. Boone, G. A. Churchill, M. O. Hengartner, T. F. C. Mackay, and D. L. Stemple (2011). The future of model organisms in human disease research. *12*.

Aittokallio, T. and B. Schwikowski (2006). Graph-based methods for analysing networks in cell biology. *Briefings in bioinformatics 7*(3), 243–255.

Altman, R. B. and T. E. Klein (2002). Challenges for biomedical informatics and pharmacogenomics. *Annual review of pharmacology and toxicology 42*(1), 113–133.

Aronson, A. R. (2001). Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, pp. 17. American Medical Informatics Association.

Arrell, D. and A. Terzic (2010). Network systems biology for drug discovery. *88*.

Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, and J. T. Eppig (2000). Gene ontology: tool for the unification of biology. *25*.

Barabasi, A., N. Gulbahce, and J. Loscalzo (2011). Network medicine: a network-based approach to human disease. *Nature Reviews Genetics 12*(1), 56–68.

Battle, A., M. Jonikas, P. Walter, J. Weissman, and D. Koller (2010). Automated identification of pathways from quantitative genetic interaction data. *Molecular Systems Biology 6*(1).

Baumgartner, W. A., K. B. Cohen, L. M. Fox, G. Acquaah-Mensah, and L. Hunter (2007). Manual curation is not sufficient for annotation of genomic databases. *Bioinformatics 23*(13), i41–i48.

Bebek, G. and J. Yang (2007). Pathfinder: mining signal transduction pathway segments from protein-protein interaction networks. *BMC bioinformatics 8*(1), 335.

Bedell, M. A., D. A. Largaespada, N. A. Jenkins, and N. G. Copeland (1997). Mouse models of human disease. part ii: recent progress and future directions. *Genes & development 11*(1), 11–43.

Berman, H., J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne (2000). The protein data bank. *Nucleic acids research 28*(1), 235–242.

Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research 32*(suppl 1), D267–D270.

Bomken, S., K. Fišer, O. Heidenreich, and J. Vormoor (2010). Understanding the cancer stem cell. *British journal of cancer 103*(4), 439–445.

BIBLIOGRAPHY

Borgwardt, K. and H. Kriegel (2005). Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on*, pp. 8 pp. IEEE.

Borgwardt, K., H. Kriegel, S. Vishwanathan, and N. Schraudolph (2007). Graph kernels for disease outcome prediction from protein-protein interaction networks. In *Proc. of Pacific Symposium on Biocomputing (PSB)*, Volume 12, pp. 4–15.

Borgwardt, K., C. Ong, S. Schnauer, S. Vishwanathan, A. Smola, and H. Kriegel (2005). Protein function prediction via graph kernels. *Bioinformatics 21*(suppl 1), i47–i56.

Borgwardt, K. M., C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel (2005). Protein function prediction via graph kernels. *Bioinformatics 21*(suppl 1), i47–i56.

Borgwardt, K. M., N. N. Schraudolph, and S. Viswanathan (2006). Fast computation of graph kernels. In *Advances in neural information processing systems*, pp. 1449–1456.

Brown, J., T. URATA, T. TAMURA, A. MIDORI, T. KAWABATA, and T. AKUTSU (2010). Compound analysis via graph kernels incorporating chirality. *Journal of Bioinformatics and Computational Biology 8*(supp01), 63–81.

Brown, P. F., V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer (1993). The mathematics of statistical machine translation: Parameter estimation. *19*.

Browne, A. C., G. Divita, A. R. Aronson, and A. T. McCray (2003). Umls language and vocabulary tools: Amia 2003 open source expo. In *AMIA Annual Symposium Proceedings*, Volume 2003, pp. 798. American Medical Informatics Association.

Butcher, E. C., E. L. Berg, and E. J. Kunkel (2004). Systems biology in drug discovery. *Nature biotechnology 22*(10), 1253–1259.

Cakmak, A. and G. Ozsoyoglu (2007). Mining biological networks for unknown pathways. *Bioinformatics 23*(20), 2775.

Campbell, K. E. and M. A. Musen (1992). Representation of clinical data using snomed iii and conceptual graphs. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pp. 354. American Medical Informatics Association.

Carbon, S., A. Ireland, C. Mungall, S. Shu, B. Marshall, and S. Lewis (2009). Amigo: online access to ontology and annotation data. *Bioinformatics 25*(2), 288–289.

Caspi, R., H. Foerster, C. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S. Rhee, A. Shearer, and C. Tissier (2008). The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic acids research 36*(suppl 1), D623–D631.

Cerami, E., E. Demir, N. Schultz, B. Taylor, and C. Sander (2010). Automated network analysis identifies core pathways in glioblastoma. *PLoS One 5*(2), e8918.

BIBLIOGRAPHY

Chang, A. A., K. M. Heskett, and T. M. Davidson (2006). Searching the literature using medical subject headings versus text word with pubmed. *The Laryngoscope 116*(2), 336–340.

Chang, C. and C. Lin (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST) 2*(3), 27.

Chautard, E., N. Thierry-Mieg, and S. Ricard-Blum (2009). Interaction networks: from protein functions to drug discovery. a review. *Pathologie Biologie 57*(4), 324–333.

Chen, L., T. Huang, X. Shi, Y. Cai, and K. Chou (2010). Analysis of protein pathway networks using hybrid properties. *Molecules 15*(11), 8177–8192.

Cogswell, J., J. Ward, I. Taylor, M. Waters, Y. Shi, B. Cannon, K. Kelnar, J. Kemppainen, D. Brown, and C. Chen (2008). Identification of mirna changes in alzheimer's disease brain and csf yields putative biomarkers and insights into disease pathways. *Journal of Alzheimer's disease 14*(1), 27–41.

Conesa, A., S. Götz, J. M. García-Gómez, J. Terol, M. Talón, and M. Robles (2005). Blast2go: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics 21*(18), 3674–3676.

Cordell, H. J. (2009). Detecting gene-gene interactions that underlie human diseases. *Nature Reviews Genetics 10*(6), 392–404.

De Bruijn, B. and J. Martin (2002). Getting to the (c) ore of knowledge: mining biomedical literature. *International journal of medical informatics 67*(1), 7–18.

Debnath, A., R. Lopez de Compadre, G. Debnath, A. Shusterman, and C. Hansch (1991). Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry 34*(2), 786–797.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–38.

DePiero, F. and D. Krout (2003). An algorithm using length-r paths to approximate subgraph isomorphism. *Pattern recognition letters 24*(1), 33–46.

Dheekollu, J. and P. M. Lieberman (2011). The replisome pausing factor timeless is required for episomal maintenance of latent epstein-barr virus. *85*.

Dwinell, M., E. Worthey, M. Shimoyama, B. Bakir-Gungor, J. DePons, S. Laulederkind, T. Lowry, R. Nigram, V. Petri, and J. Smith (2009). The rat genome database 2009: variation, ontologies and pathways. *Nucleic acids research 37*(suppl 1), D744–D749.

Erdmann, M., A. Maedche, H.-P. Schnurr, and S. Staab (2000). From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In *Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content*, pp. 79–85. Association for Computational Linguistics.

Espinosa, G., D. Yaffe, Y. Cohen, A. Arenas, and F. Giralt (2000). Neural network based quantitative structural property relations (qsprs) for predicting boiling points of aliphatic hydrocarbons. *Journal of Chemical Information and Computer Sciences 40*(3), 859–879.

Faro, A., D. Giordano, and C. Spampinato (2012). Combining literature text mining with microarray data: advances for system biology modeling. *Briefings in bioinformatics 13*(1), 61–82.

Fei, H. and J. Huan (2008). Structure feature selection for graph classification. pp. 991–1000. ACM.

Feitsma, H. and E. Cuppen (2008). Zebrafish as a cancer model. *6*.

Felsenstein, J. (2004). *Inferring phylogenies*. Sinauer Associates.

Ferrer, L., A. G. Shearer, and P. D. Karp (2011). Discovering novel subsystems using comparative genomics. *27*.

Floyd, R. (1962). Algorithm 97: shortest path. *Communications of the ACM 5*(6), 345.

Franke, L., H. Bakel, L. Fokkens, E. De Jong, M. Egmont-Petersen, and C. Wijmenga (2006). Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *The American Journal of Human Genetics 78*(6), 1011–1025.

Gärtner, T., P. Flach, and S. Wrobel (2003). On graph kernels: Hardness results and efficient alternatives. *Learning Theory and Kernel Machines*, 129–143.

Glass, C. K., K. Saijo, B. Winner, M. C. Marchetto, and F. H. Gage (2010). Mechanisms underlying inflammation in neurodegeneration. *Cell 140*(6), 918–934.

Goto, N., P. Prins, M. Nakao, R. Bonnal, J. Aerts, and T. Katayama (2010). Bioruby: Bioinformatics software for the ruby programming language. *Bioinformatics 26*(20), 2617.

Gudes, E., S. E. Shimony, and N. Vanetik (2006). Discovering frequent graph patterns using disjoint paths. *18*.

Guldener, U., M. Munsterkotter, M. Oesterheld, P. Pagel, A. Ruepp, H.-W. Mewes, and V. Stumpflen (2006). Mpact: the mips protein interaction resource on yeast. *Nucleic acids research 34*(suppl 1), D436–D441.

BIBLIOGRAPHY

Haldar, M., J. D. Hancock, C. M. Coffin, S. L. Lessnick, and M. R. Capecchi (2007). A conditional mouse model of synovial sarcoma: insights into a myogenic origin. *Cancer cell 11*(4), 375–388.

Haldar, M., R. L. Randall, and M. R. Capecchi (2008). Synovial sarcoma: From genetics to genetic-based animal modeling. *Clinical Orthopaedics and Related Research 466*, 2156–2167.

Hall, M. (1999). *Correlation-based feature selection for machine learning*. Ph. D. thesis.

Hamosh, A., A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick (2005). Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic acids research 33*(suppl 1), D514–D517.

Harchaoui, Z. and F. Bach (2007). Image classification with segmentation graph kernels. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE.

Harris, T. W., I. Antoshechkin, T. Bieri, D. Blasiar, J. Chan, W. J. Chen, N. De La Cruz, P. Davis, M. Duesbury, and R. Fang (2010). Wormbase: a comprehensive resource for nematode research. *Nucleic acids research 38*(suppl 1), D463–D467.

Helma, C., R. King, S. Kramer, and A. Srinivasan (2001). The predictive toxicology challenge 2000äì2001. *Bioinformatics 17*(1), 107–108.

Hennessy, B. T., D. L. Smith, P. T. Ram, Y. Lu, and G. B. Mills (2005). Exploiting the pi3k/akt pathway for cancer drug discovery. *Nature Reviews Drug Discovery 4*(12), 988–1004.

Hofmann, T., B. Schölkopf, and A. J. Smola (2008). Kernel methods in machine learning. *The annals of statistics*, 1171–1220.

Horv °th, T., T. G§rtner, and S. Wrobel (2004). Cyclic pattern kernels for predictive graph mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 158–167. ACM.

Hu, H., X. Yan, Y. Huang, J. Han, and X. Zhou (2005). Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics 21*(suppl 1), i213.

Huang, T., L. Chen, Y.-D. Cai, and K.-C. Chou (2011). Classification and analysis of regulatory pathways using graph property, biochemical and physicochemical property, and functional property. *6*.

Ito, T., T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki (2001). A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences 98*(8), 4569–4574.

Jensen, P. B., L. J. Jensen, and S. Brunak (2012). Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics 13*(6), 395–405.

Jin, N., C. Young, and W. Wang (2009). Graph classification based on pattern co-occurrence. pp. 573–582. ACM.

Jin, N., C. Young, and W. Wang (2010). Gaia: graph classification using evolutionary computation. pp. 879–890. ACM.

Joshi-Tope, G., M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, and L. Matthews (2005). Reactome: a knowledgebase of biological pathways. *33*.

Kanehisa, M., S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa (2010). Kegg for representation and analysis of molecular networks involving diseases and drugs. *38*.

Karathia, H., E. Vilaprinyo, A. Sorribas, and R. Alves (2011). Saccharomyces cerevisiae as a model organism: a comparative study. *PloS one 6*(2), e16015.

Karnovsky, A., T. Weymouth, T. Hull, G. Tarcea, G. Scardoni, C. Laudanna, M. Sartor, K. Stringer, H. V. Jagadish, C. Burant, B. Athey, and G. Omenn (2012). Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data. *Bioinformatics 28*(3), 373–380.

Karp, P., M. Riley, S. Paley, and A. Pellegrini-Toole (2002). The metacyc database. *Nucleic acids research 30*(1), 59–61.

Kashima, H., K. Tsuda, and A. Inokuchi (2003). Marginalized kernels between labeled graphs. In *ICML*, Volume 3, pp. 321–328.

Kerrien, S., Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, and R. Huntley (2007). Intact - open source resource for molecular interaction data. *Nucleic acids research 35*(suppl 1), D561–D565.

Khatri, P., M. Sirota, and A. Butte (2012). Ten years of pathway analysis: Current approaches and outstanding challenges. *PLoS Computational Biology 8*(2), e1002375.

Kipper, K., A. Korhonen, N. Ryant, and M. Palmer (2006). Extending verbnet with novel verb classes. In *Proceedings of LREC*, Volume 2006, pp. 1.

Kondor, R. I. and J. Lafferty (2002). Diffusion kernels on graphs and other discrete input spaces. In *ICML*, Volume 2, pp. 315–322.

Kong, X., W. Fan, and P. S. Yu (2011). Dual active feature and sample selection for graph classification. ACM.

Koteja, P., T. Garland, J. K. Sax, J. G. Swallow, and P. A. Carter (1999). Behaviour of house mice artificially selected for high levels of voluntary wheel running. *Animal behaviour 58*(6), 1307–1318.

BIBLIOGRAPHY

Kulis, B., S. Basu, I. Dhillon, and R. Mooney (2009). Semi-supervised graph clustering: a kernel approach. *74*.

Lagoze, C., S. Payette, E. Shin, and C. Wilper (2006). Fedora: an architecture for complex objects and their relationships. *International Journal on Digital Libraries 6*(2), 124–138.

Lai, L., A. Liberzon, J. Hennessey, G. Jiang, J. Qi, J. Mesirov, and X. Steven (2012). Arapath: a knowledgebase for pathway analysis in arabidopsis. *Bioinformatics 28*(17), 2291–2292.

Lambert, J., B. Grenier-Boley, V. Chouraki, S. Heath, D. Zelenika, N. Fievet, D. Hannequin, F. Pasquier, O. Hanon, and A. Brice (2010). Implication of the immune system in alzheimer's disease: evidence from genome-wide pathway analysis. *Journal of Alzheimer's disease 20*(4), 1107–1118.

Lehne, B. and T. Schlitt (2009). Protein-protein interaction databases: Keeping up with growing interactomes. *Human genomics 3*(3), 291–297.

Leskovec, J., M. Grobelnik, and N. Milic-Frayling (2004). Learning sub-structures of document semantic graphs for document summarization.

Li, E. and K. Hristova (2006). Role of receptor tyrosine kinase transmembrane domains in cell signaling and human pathologies. *Biochemistry 45*(20), 6241–6251.

Li, G., M. Semerci, B. Yener, and M. Zaki (2011). Graph classification via topological and label attributes. In *9th Workshop on Mining and Learning with Graphs (with SIGKDD)*.

Lin, J., C. M. Gan, X. Zhang, S. Jones, T. Sjblom, L. D. Wood, D. W. Parsons, N. Papadopoulos, K. W. Kinzler, and B. Vogelstein (2007). A multidimensional analysis of genes mutated in breast and colorectal cancers. *Genome research 17*(9), 1304–1318.

Lipscomb, C. E. (2000). Medical subject headings (mesh). *Bulletin of the Medical Library Association 88*(3), 265.

Liu, G., L. Wong, and H. Chua (2009). Complex discovery from weighted ppi networks. *Bioinformatics 25*(15), 1891.

Liu, N. and E. N. Olson (2010). Microrna regulatory networks in cardiovascular development. *18*.

Marcotte, E. M., M. Pellegrini, H.-L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg (1999). Detecting protein function and protein-protein interactions from genome sequences. *Science 285*(5428), 751–753.

Maudsley, S., W. Chadwick, L. Wang, Y. Zhou, B. Martin, and S. Park (2011). Bioinformatic approaches to metabolic pathways analysis. *Methods in molecular biology (Clifton, NJ) 756*, 99.

McDowall, M. D., M. S. Scott, and G. J. Barton (2009). Pips: human protein-protein interaction prediction database. *Nucleic acids research 37*(suppl 1), D651–D656.

McGary, K. L., T. J. Park, J. O. Woods, H. J. Cha, J. B. Wallingford, and E. M. Marcotte (2010). Systematic discovery of nonobvious human disease models through orthologous phenotypes. *Proceedings of the National Academy of Sciences 107*(14), 6544–6549.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM 38*(11), 39–41.

Moon, T. K. (1996). The expectation-maximization algorithm. *13*.

Morrison, A., R. Christensen, J. Alley, A. Beck, E. Bernstine, J. Lemontt, and C. Lawrence (1989). Rev3, a saccharomyces cerevisiae gene whose function is required for induced mutagenesis, is predicted to encode a nonessential dna polymerase. *171*.

Muller, K., S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks 12*(2), 181–201.

Myers, C. L. and O. G. Troyanskaya (2007). Context-sensitive data integration and prediction of biological networks. *Bioinformatics 23*(17), 2322–2330.

Nabhan, A. R. and I. N. Sarkar (2012). Mining disease fingerprints from within genetic pathways. In *AMIA Annual Symposium Proceedings*, Volume 2012, pp. 1320. American Medical Informatics Association.

Nabhan, A. R. and I. N. Sarkar (2013). Structural network analysis of biological networks for assessment of potential disease model organisms. *Journal of Biomedical Informatics*.

Nijssen, S. and J. Kok (2004). A quickstart in frequent structure mining can make a difference. pp. 647–652. ACM.

Novoyatleva, T., F. Diehl, M. J. Van Amerongen, C. Patra, F. Ferrazzi, R. Bellazzi, and F. B. Engel (2010). Tweak is a positive regulator of cardiomyocyte proliferation. *85*.

Osterman, A. and R. Overbeek (2003). Missing genes in metabolic pathways: a comparative genomics approach. *Current opinion in chemical biology 7*(2), 238–251.

Pan, T., S. Kondo, W. Le, and J. Jankovic (2008). The role of autophagy-lysosome pathway in neurodegeneration associated with parkinson's disease. *Brain 131*(8), 1969.

Pawson, T. and R. Linding (2008). Network medicine. *582*.

Peri, S., J. D. Navarro, R. Amanchy, T. Z. Kristiansen, C. K. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, T. Gandhi, and M. Gronborg (2003). Development of human protein reference database as an initial platform for approaching systems biology in humans. *13*.

Pico, A., T. Kelder, M. Van Iersel, K. Hanspers, B. Conklin, and C. Evelo (2008). Wikipathways: pathway editing for the people. *PLoS biology 6*(7), e184.

Prasad, T., R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, and A. Venugopal (2009). Human protein reference database - 2009 update. *Nucleic acids research 37*(suppl 1), D767–D772.

Ralaivola, L., S. Swamidass, H. Saigo, and P. Baldi (2005a). Graph kernels for chemical informatics. *Neural Networks 18*(8), 1093–1110.

Ralaivola, L., S. J. Swamidass, H. Saigo, and P. Baldi (2005b). Graph kernels for chemical informatics. *Neural Networks 18*(8), 1093–1110.

Ramon, J. and T. Gärtner (2003). Expressivity versus efficiency of graph kernels. In *First International Workshop on Mining Graphs, Trees and Sequences*, pp. 65–74.

Ranu, S. and A. K. Singh (2009). Graphsig: A scalable approach to mining significant subgraphs in large graph databases. IEEE.

Read, R. and D. Corneil (1977). The graph isomorphism disease. *Journal of Graph Theory 1*(4), 339–363.

Rebholz-Schuhmann, D., A. Oellrich, and R. Hoehndorf (2012). Text-mining solutions for biomedical research: enabling integrative biology. *Nature Reviews Genetics*.

Rual, J., K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. Berriz, F. Gibbons, M. Dreze, and N. Ayivi-Guedehoussou (2005). Towards a proteome-scale map of the human protein-protein interaction network. *Nature 437*(7062), 1173–1178.

Rudy, Y., M. J. Ackerman, D. M. Bers, C. E. Clancy, S. R. Houser, B. London, A. D. McCulloch, D. A. Przywara, R. L. Rasmusson, and R. J. Solaro (2008). Systems approach to understanding electromechanical activity in the human heart. *118*.

Saigo, H., S. Nowozin, T. Kadowaki, T. Kudo, and K. Tsuda (2009). gboost: a mathematical programming approach to graph classification and regression. *Machine learning 75*(1), 69–89.

Sandelin, A., W. W. Wasserman, and B. Lenhard (2004). Consite: web-based prediction of regulatory elements using cross-species comparison. *Nucleic acids research 32*(suppl 2), W249–W252.

Schilling, C. H., D. Letscher, and B. . Palsson (2000). Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of theoretical biology 203*(3), 229–248.

Seeland, M., T. Girschick, F. Buchwald, and S. Kramer (2010). Online structural graph clustering using frequent subgraph mining. *Machine Learning and Knowledge Discovery in Databases*, 213–228.

Senf, A. and X.-w. Chen (2009). Identification of genes involved in the same pathways using a hidden markov model-based approach. *Bioinformatics 25*(22), 2945–2954.

Sengupta, D. J., B. Zhang, B. Kraemer, P. Pochart, S. Fields, and M. Wickens (1996). A three-hybrid system to detect rna-protein interactions in vivo. *Proceedings of the National Academy of Sciences 93*(16), 8496–8501.

Shang, H., Y. Zhang, X. Lin, and J. X. Yu (2008). Taming verification hardness: an efficient algorithm for testing subgraph isomorphism. *1*.

Shatkay, H. and R. Feldman (2003). Mining the biomedical literature in the genomic era: an overview. *Journal of computational biology 10*(6), 821–855.

Shervashidze, N. and K. M. Borgwardt (2009). Fast subtree kernels on graphs. In *Advances in Neural Information Processing Systems*, pp. 1660–1668.

Shervashidze, N., S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt (2009). Efficient graphlet kernels for large graph comparison. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics. Society for Artificial Intelligence and Statistics*.

Shokoufandeh, A., S. Dickinson, K. Siddiqi, and S. Zucker (1999). Indexing using a spectral encoding of topological structure. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, Volume 2. IEEE.

Slattery, M., R. Wolff, K. Curtin, F. Fitzpatrick, J. Herrick, J. Potter, B. Caan, and W. Samowitz (2009). Colon tumor mutations and epigenetic changes associated with genetic polymorphism: Insight into disease pathways. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis 660*(1-2), 12–21.

Sowa, J. F. (2008). Conceptual graphs. *Foundations of Artificial Intelligence 3*, 213–237.

Stark, C., B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers (2006). Biogrid: a general repository for interaction datasets. *Nucleic acids research 34*(suppl 1), D535–D539.

Stelzl, U., U. Worm, M. Lalowski, C. Haenig, F. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, and S. Koeppen (2005). A human protein-protein interaction network: a resource for annotating the proteome. *Cell 122*(6), 957–968.

Stern, H. M. and L. I. Zon (2003). Cancer genetics and drug discovery in the zebrafish. *Nature Reviews Cancer 3*(7), 533–539.

Stoletov, K. and R. Klemke (2008). Catch of the day: zebrafish as a human cancer model. *Oncogene 27*(33), 4509–4520.

Sullivan, C. and C. H. Kim (2008). Zebrafish as a model for infectious disease and immune function. *25*.

Szklarczyk, D., A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, and P. Bork (2011). The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *39*.

Tan, P. N., V. Kumar, and J. Srivastava (2002). Selecting the right interestingness measure for association patterns. ACM.

Thomas, M. A., L. Yang, B. J. Carter, and R. D. Klaper (2011). Gene set enrichment analysis of microarray data from pimephales promelas (rafinesque), a non-mammalian model organism. *12*.

Tian, L., S. A. Greenberg, S. W. Kong, J. Altschuler, I. S. Kohane, and P. J. Park (2005). Discovering statistically significant pathways in expression profiling studies. *102*.

Tsuda, K., T. Kin, and K. Asai (2002). Marginalized kernels for biological sequences. *Bioinformatics 18*(suppl 1), S268–S275.

Tweedie, S., M. Ashburner, K. Falls, P. Leyland, P. McQuilton, S. Marygold, G. Millburn, D. Osumi-Sutherland, A. Schroeder, and R. Seal (2009). Flybase: enhancing drosophila gene ontology annotations. *Nucleic acids research 37*(suppl 1), D555–D559.

Twigger, S. N., M. Shimoyama, S. Bromberg, A. E. Kwitek, and H. J. Jacob (2007). The rat genome database, update 2007 - easing the path from disease to data and back again. *Nucleic acids research 35*(suppl 1), D658–D662.

Vishwanathan, S., N. N. Schraudolph, R. Kondor, and K. M. Borgwardt (2010). Graph kernels. *The Journal of Machine Learning Research 99*, 1201–1242.

Vogelstein, B. and K. W. Kinzler (2004). Cancer genes and the pathways they control. *Nature medicine 10*(8), 789–799.

Vogelstein, J., W. Gray, R. Vogelstein, and C. Priebe (2011). Graph classification using signal subgraphs: Applications in statistical connectomics. *Arxiv preprint arXiv:1108.1427*.

Wale, N. and G. Karypis (2006). Comparison of descriptor spaces for chemical compound retrieval and classification. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pp. 678–689. IEEE.

Walhout, A. J. and M. Vidal (2001). Protein interaction maps for model organisms. *Nature Reviews Molecular Cell Biology 2*(1), 55–63.

Wang, G., B. Wang, X. Yang, and G. Yu (2012). Efficiently indexing large sparse graphs for similarity search. *Knowledge and Data Engineering, IEEE Transactions on* (99), 1–1.

Whelan, R. S., V. Kaplinskiy, and R. N. Kitsis (2010). Cell death in the pathogenesis of heart disease: mechanisms and significance. *Annual review of physiology 72*, 19–44.

Wolf, Y. I., I. B. Rogozin, A. S. Kondrashov, and E. V. Koonin (2001). Genome alignment, evolution of prokaryotic genome organization, and prediction of gene function using genomic context. *Genome Research 11*(3), 356–372.

Yan, X., H. Cheng, J. Han, and P. Yu (2008). Mining significant graph patterns by leap search. pp. 433–444. ACM.

Yan, X. and J. Han (2002). gspan: Graph-based substructure pattern mining. IEEE.

Yan, X., P. Yu, and J. Han (2004). Graph indexing: a frequent structure-based approach. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 335–346. ACM.

Yoo, I., X. Hu, and I.-Y. Song (2007). A coherent graph-based semantic clustering and summarization approach for biomedical literature and a new summarization evaluation method. *BMC bioinformatics 8*(Suppl 9), S4.

You, C., L. Holder, and D. Cook (2009). Substructure analysis of metabolic pathways by graph-based relational learning.

Yu, L. and H. Liu (2004). Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research 5*, 1205–1224.

Zhao, P., J. Yu, and P. Yu (2007). Graph indexing: tree+ delta¡= graph. In *Proceedings of the 33rd international conference on Very large data bases*, pp. 938–949. VLDB Endowment.

Zheleva, E. and L. Getoor (2008). Preserving the privacy of sensitive relationships in graph data. In *Privacy, security, and trust in KDD*, pp. 153–171. Springer.

Zhernakova, A., C. C. van Diemen, and C. Wijmenga (2009). Detecting shared pathogenesis from the shared genetics of immune-related diseases. *Nature Reviews Genetics 10*(1), 43–55.

Zon, L. I. and R. T. Peterson (2005). In vivo drug discovery in the zebrafish. *4*.