# APPLICATION OF GRAPH BASED DATA MINING TO BIOLOGICAL NETWORKS

by

CHANG HUN YOU

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2005

## ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervising professor, Dr. Lawrence Holder, for his patience, encouragement, and expert advice. My first research and thesis could not have been finished without his guidance and belief.

I would like to thank Dr. Diane Cook for her first leading to the Artificial Intelligence world, reviewing and guiding my work. I would also to thank Dr. Lynn Peterson for guidance as my thesis committee member.

I want to thank my friends and colleagues for their valuable advice and encouragement.

I would like to express my appreciation to my family. My father has motivated and encouraged me through his life. My mother has always supported and believed in me. I appreciate my sister and brother for their support and understanding. My love, Hyein Nam, has always been my soul and power behind me, and is specially appreciated.

November 18, 2003

## ABSTRACT

APPLICATION OF GRAPH BASED DATA MINING
TO BIOLOGICAL NETWORKS

Publication No. _____

CHANG HUN YOU, MS
The University of Texas at Arlington, 2005

Supervising Professor: Lawrence B. Holder

A huge amount of biological data has been generated by long-term research. It is time to start to focus on a system-level understanding of bio-systems. Biological networks are networks of biochemical reactions, containing various objects and their relationships. Understanding of biological networks is a starting point of systems biology.

Multi-relational data mining finds the relational patterns in both the entity attributes and relations in the data. A widely used representation for relational data is a graph consisting of vertices and edges between these vertices. Graph-based data mining, as one approach of multi-relational data mining, finds relational patterns in a graph representation of data.

This thesis will present a graph representation of biological networks including almost all features of pathways, and apply the Subdue graph-based data mining system in both supervised and unsupervised settings. This research will also show that the patterns found by Subdue have important biological meaning.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

When Watson and Crick found the structure of nucleic acids, they defined the results by making hand models with small plastic balls and wires [4]. For more than 60 years research has mainly focused on genomics and proteomics. After accumulating many kinds of results regarding to genomics and proteomics and completing the Human Genome Project, many biologists and computational biologists are focused on several new challenges. One of the open problems is systems biology, which gives us a system-level understanding of bio-systems. It is finally time to study their systems more comprehensively based on results of 60 years. One of the main challenges of systems biology is understanding biological networks. The biological network is a network of biochemical compounds, proteins, other gene products and their relationships. As the Internet as the network of networks has played a central role in computer science in the latter of 20th century, the biological network will play a major role in post-genomic bioinformatics researche

Several efforts on the frontier of systems biology have been resulted in significant achievement in genomics and proteomics [1]. Knowledge discovery in existed biological networks should be a good resource for modeling unrecognized biological networks. A graph has been used as a popular data structure to represent a wide variety of relational data such as computer networks, social networks, and biological data [5]. A biological network is another field to be represented as a graph.

The Subdue graph-based data mining system has been successfully applied to various areas such as security [6], web search [7] and protein structures [8]. In this research,

a graph representation of biological networks is generated from the KEGG PATHWAY database. The KEGG PATHWAY database is one of the major repositories of biological networks. It has a standard file format, KGML, to distribute biological network information. KGML defines objects of the biological network and their relationships as an XML data structure. It had 271 species and 167 reference pathways in August 2005, and is updated continuously. The first step of this research is converting KGML data to graph form to be recognized by the Subdue.

After completing the graph representation step, the graph of biological networks comprises several sets for experiments as a category of biological networks or a category of species. Subdue is applied as two typical approaches of knowledge discovery: supervised and unsupervised. In supervised pattern learning, Subdue tries to find patterns to distinguish two sets of biological networks. Two supervised mining experiments are presented. The first distinguishes between one network of a group of some species and another network of same group. The second approach tries to find patterns to distinguish between a group of biological networks of one species and the same group of biological networks of another species. Subdue can performs well on the first the first approaches, but cannot distinguish well the sets in the second approach. It is the reason that biological networks have few species-specific feature even though proteins and genes have many species-specific features. Species-specific means pertaining to or limited to one species, but not to general. In unsupervised pattern learning, Subdue is looking for common patterns. First, it tries to find common substructures in one kind of biological network across a group of species. Second, it is looking for patterns in a group of networks for one species.

Chapter 2 provides a brief introduction to data mining and bioinformatics. Then, multi-relational data mining is introduced along with some multi-relation data mining approaches to bioinformatics. Chapter 3 describes the Subdue a graph-based data mining

system. The main algorithms of Subdue are introduced along with several concepts. Chapter 4 describes the biological networks data used in this research. First, essential concepts of biochemistry and systems biology are explained. Then, the chapter will give an overview of biological networks and databases of biological networks. Lastly we explain the KEGG PATHWAY database and KGML KEGG Markup Language as a way of representing and distributing the biological networks data. Chapter 5 presents the main experiments. A graph representation of biological networks will be introduced. Then supervised pattern learning and unsupervised pattern learning will be applied as the main experiments. The results are presented and discussed in Chapter 6. Chapter 7 presents conclusions and future work.

# CHAPTER 2

# MULTI-RELATIONAL DATA MINING AND BIOINFORMATICS

In this chapter we provide a survey of related works. First, a brief introduction of Data Mining will be provided. Then fundamental research areas in bioinformatics will be described. Lastly we describe Multi-Relation Data Mining (MRDM) and its approach to the biological domain. Logic-based data mining and graph-based data mining will be given as examples of multi-relational data mining.

## 2.1 Data Mining

Data mining, which is also referred to as knowledge discovery in databases, is a process of extraction of previously unknown and potentially useful information from large databases or information archives [9, 10]. Mining data and knowledge from databases has been a key research topic. The knowledge discovery process consists of an iterative sequence of the following steps [10]: Data cleaning (to handle noise and inconsistent data), Data integration (to combine multiple, heterogeneous data sources), Data selection (to retrieve relevant data from the database), Data transformation (to transform the data into the specific format for data mining), Data mining (to find interesting and meaning patterns in the data), Pattern evaluation (to evaluate the patterns along with the reasonable measure technique), and Knowledge presentation (to present discovered knowledge to the user).

Data mining techniques can be classified according to different views such as what kinds of knowledge to work on, what kinds of databases to be mined, and what kinds of algorithms to be applied [9].

This section categorizes data mining based on algorithms to be applied.

*Mining association rules* [9, 10] finds association rules like $A_1 \wedge A_2 \Rightarrow B_1 \wedge B_2$ from the relevant data sets in a database. It is a example of association rule that the buyer ($A_1$) of beer ($A_2$) will buy ($B_1$) peanuts ($B_2$), too. This association rule describes the relationships among data in a given set.

*Data generalization* [9, 10] gives an understandable description of a large database. using abstraction, summarization and characterization. A sentence, "One who buys beer will buy peanuts, too.", is transformed to "$Buy(x) \wedge Is(x, beer) \Rightarrow Buy(y) \wedge Is(y, peanuts)$ as simple logic form. This transformation is an example of data generalization. On-line analytical processing (OLAP) and data warehousing are also techniques for data generalization.

*Mining classification rules* [9, 10] distinguishes a large test set of data into several groups based on classification rules generated in the training set. The first step learns classification rules in the training sets. The second step applies these classification rules to the test set to classify them into each group. This approach is also called supervised learning, because of the presence of a pre-classified training set.

*Data clustering* [9, 10] groups a set of data into clusters without any predefined rules. This approach tries to maximize similarity in the same cluster and minimize similarity between different clusters. This algorithm is called unsupervised learning in contrast to supervised learning.

We described typical data mining algorithms briefly. There are several complex types of databases such as a temporal databases, spatial database, multimedia database and text database [10]. The domains of data mining have become broad such as business, homeland security and biology domain. Each domain requires an appropriate data mining algorithm.

## 2.2 Bioinformatics

Bioinformatics is the application of computational techniques to analyze the information associated with bio-molecules [1, 11]. The huge amount of biological data provides many challenges to researchers. Bioinformatics can be defined by four words: Information, Organization, Analysis and Application [1]. This section consists of two subsections. The first, information and organization, will introduce a variety of biological information and its organization. The second, analysis and application, will describe the ways to analyze data and their application.

### 2.2.1 Information and organization

Table 2.1. A variety of data used in bioinformatics

| Data | Data amount | Bioinformatics Topic |
|---|---|---|
| DNA sequence | 51,674,486,881 bases<br>in 46,947,388 sequence<br>in GenBank at NCBI | Sequence Alignment<br>Separating introns and exon<br>Phylogenetic Prediction |
| Protein Sequence | 195,589 sequence entries,<br>70852380 amino acids<br>abstracted from 134,391 references<br>in the UniProtKB/Swiss-Prot | Sequence Alignment or MSA<br>equence Alignment or MSA<br>Protein Sequence Prediction |
| Protein Structure | 33,065 Structures | Protein Structure Prediction<br>Protein Function Prediction |
| Pathway | 30,224 pathways generated<br>from 246 reference pathways | Biological Network Modeling<br>Systems Biology |

Long-time cumulative biological research has generated various kinds and a huge amount of data. GenBank, the largest gene sequence database, has 46,947,388 sequences of genes, and UniProt, the largest protein database, has 195,589 sequences of proteins. Table 2.1 shows a variety of the huge amount of biological data. It has become impossible to analyze those data by the hands of biologists. There are many kinds of data for

biological research such as raw DNA sequences, Protein sequences, Protein structure, Genome data, Pathway data, Disease and Gene Expression data. These days literature data which are databases of references for research become an important resource [11]. It is necessary to employ reliable and efficient methods to maintain this data because of its huge amount and various kinds.

Table 2.2. Databases of Biological Information

| Database Type | Examples |
|---|---|
| Nucleotide Sequence | GenBank *http://www.ncbi.nlm.nih.gov*<br>DDBJ : http://www.ddbj.nig.ac.jp<br>EMBL : http://www.embl.org |
| Protein Sequence | UNI-PROT *http://www.ebi.ac.uk/uniprot/*<br>PROSITE *http://au.expasy.org/prosite/* |
| Protein Structure | PDB *http://www.rcsb.org/pdb/* |
| Biomedical Literature | PubMed *http://www.ncbi.nlm.nih.gov/entrez/query.fcgi*<br>Distributed Annotation System *http://stein.cshl.org/das/* |
| Molecular Disease | OMIM *http://www.ncbi.nlm.nih.gov/Omim/* |
| Gene Expression | GEO *http://www.ncbi.nlm.nih.gov/geo/* |

The biological data also has some special features: redundancy and multiplicity [1]. An organism may have a huge amount of genes. Different gene sequences may have the same structure or a single gene may have multiple functions. Also, many sequences of genes and proteins give us redundant data. The simple store of biological data is able to give few help to researchers. Therefore, organization of data is an indispensable issue [1, 11, 12, 13]. Organizing biological data is a fundamental starting point of bioinformatics research. It allows researchers to access existing information based on their features and to submit new entries as they are produced by following the rules of database. From the early days of bioinformatics many computer scientists and biologists have been focusing on organizing and managing their data for future research, not simply storing. There are

some examples of databases which contain biological data for this purposes in table 2.2 [1]. GenBank, EMBL and DDBJ databases contain DNA sequences used for transcription to RNA sequences [14, 15, 16]. UniProt is the most comprehensive database of the protein sequences which are translated from RNA sequences [17]. PROSITE, a database of protein families and domains, contains biologically significant sites, patterns and profiles for identification of protein families [18]. PDB, the Protein Data Bank, is a primary database of 3D structures for macromolecules such as proteins, RNA, DNA and various complexes [19]. PubMed, a web achieve of the National Library of Medicine, contains links to the 15 million citations or other resources from most of life science journals for biomedical articles from the 1950s to now [20]. DAS, Distributed Annotation System, is a web service to exchang annotations on genomic sequence data [21]. GEO, Gene Expression Omnibus, is a comprehensive repository of a gene expression including a curated, online resource for gene expression data [22].

### 2.2.2 Analysis and application

The next step of bioinformatics is to understand bioinformatics data which are organized well, and interpret and apply this knowledge in a biological meaningful manner.

To analysze biological data, various experimental techniques and analysis tools have been developed. Pairwise Sequence Alignment is a basic algorithm to analysze gene and protein data [23, 24, 25]. The algorithm of alignment has been used widely in BLAST [26, 27], ClustalW [28] and MAS (Multiple Sequence Alignment) [29]. Hidden Markov Models (HMM) are used in protein family studies, identification of protein structural motifs, and gene structure prediction [30].

Due to efficient analysis methods, a variety of research has bee pursued. Transcription regulation is the research for understanding all aspects of genetic activity by analysis of DNA-binding proteins and other transcription factors. Structural studies of

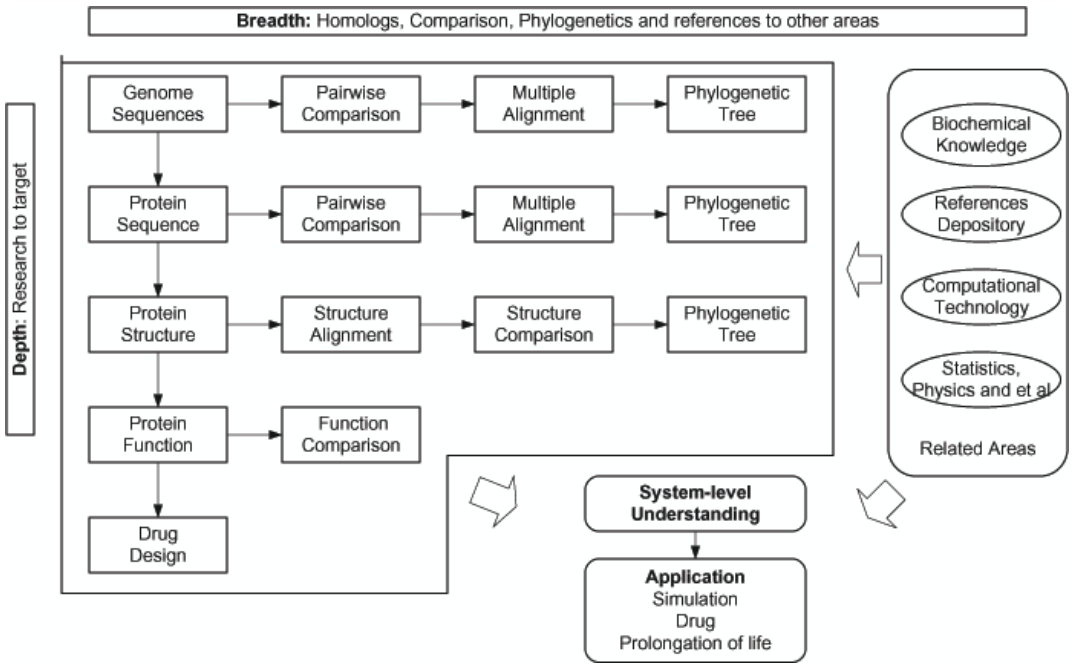



Figure 2.1. Bioinformatics spectrum: An expansion of biological research in breadth and depth [1].

macro molecules, especially proteins, are important to understand their functions by analysis of binding with other molecules, structural taxonomy and the relation between functions and structures [1]. Microarray analysis [31, 32, 33] has accumulated biological data to study. Systems Biology is focusing on system-level understanding based on genomic and proteomic results. The study of biological networks, which are the networks of biological reactions, is the main approach of systems biology [34].

The research by Luscombe and et al [1] introduced the bioinformatics spectrum to describe the research ares of bioinformatics. We used the new bioinformatics spectrum which is updated to broad areas of bioinformatics to describe the application of bioinformatics. The spectrum has two dimensions: depth and breadth. [1]. The depth can be explained with possible approaches to the target like the drug design. This axis shows that starting with gene sequences analysis. Gene sequences analysis results in

identification of protein sequences. Protein structure can be predicted from there, and protein function can be predicted based on its structure. Using this result along with resources from other areas such as biochemistry, statistics, physics, computer science and literature information, one can design a drug that specifically alters the protein's function. The breadth in biological analysis is from to compare a gene or protein with others to construct phylogenetic trees. A phylogenetic tree provides a way to evolutionarily compare two or more bio-organisms. Incorporation of a breadth study to compare other genes, molecules and organisms, and depth research of target molecule including sequence, structure, binding and altered materials, will provide a variety of knowledge to understand bio-organisms along with references of other areas. This result allows us to understand bio-systems at the system level. Finally system-level understanding will give us the efficient application such as simulation of bio-systems, production of more specific and utilizable drug, and one of the ultimate goal, prolongation of life.

## 2.3 Mulit-Relational Data Mining to Bioinformatics

### 2.3.1 Overview of Mulit-Relational Data Mining

Typical data mining approach focus on the single relational data. A variety of data in the multiple relations are provided and need to be analyzed like the biological data. Multi-relational data mining is the knowledge discovery technique in the multiple relations. Multi-relational data mining is focused on not only data in the multiple tables but also their relationships. First step of multi-relational data mining is to represent the data along with its multiple relations. First-order logics and graph representations are used for representation of multi-relational data. They used their rules to describe the data and its relationship. Mutli-relational data mining algorithms find a interesting associated rule in the representation for better understanding of multi-relational data.

### 2.3.2 Multi-Relational Data Mining Approaches to Bioinformatics

Biological database not only have a huge amount of data but also multiple relations. Therefore multi-relational data mining approaches are necessary to mine their data. For a long time many biologists have doubted that only genetics and molecular biology could solve the main problems in the biology. Biology is not just logic and engineering. Biology is active and dynamic within multiple environmental conditions. If once a drug is created from long time research, it could not be applied to humans directly. We still do not know its side effect in vivo (living organism) with various conditions of patients such as food, age, sex, constitution, climate and interaction with another drug. Therefore biological data are more complex and systemic than we expect, and they require multi-relational data mining methods.

There are several bioinformatics areas to apply MRDM approaches such as structural biology, literature discovery and biological networks. The common feature is that they are constructed from multiple data types and their relations. It is necessary to employ an efficient representation method of biological data and proper data mining techniques for knowledge discovery.

First-order logics are one of the widely used representation method in multi-relational data mining approaches [35]. Logic-based data mining, also called Inductive Logic Programming (ILP), represents data using logic. ILP is generally used in biological data [12]. Observed clauses and background knowledge are combined by using the ILP system to generate resultant rule. Then the system can distinguish between positive examples covered by the resultant rule, and negative examples [36]. Support-Vector ILP (SVILP) using Support Vector Machines and ILP provides a new approach which not only captures the semantic and syntactic relation in the data but also gives the flexibility of using arbitrary forms of structured and non-structured data coded in a relational method [37].

SVILP is applied successfully in the systems biology area. Stochastic Logic Programs (SLP) provide an efficient representation for metabolic pathways [38].

Graph is a pervasive data structure and widely used in a variety of areas. A lot of biological problems are represented and solved by using graph such as DNA sequencing and protein identification [39]. The next chapter will describe graph-based data mining which is the focus of this research.

## 2.4 Summary

This chapter explains related works. We introduced briefly the concept of data mining and its algorithms. Then, an introduction to bioinformatics was given along with organization of biological data, and analysis and application of this data. Lastly we described multi-relational data mining approaches to bioinformatics along with logic-based data mining. The next chapter will contain graph-based data mining as another approach to multi-relational data mining.

# CHAPTER 3

# GRAPH-BASED DATA MINING

## 3.1 Overview of Graph-Based Data Mining

Bioinformatics domains have a variety of structural data such as genomes, proteins and biological networks. One of the most common ways of describing structural data is a graph representation. The graph is an abstract data structure consisting of vertices and edges which are relationship between vertices [39].

Graph-based data mining denotes a collection of algorithms for mining the relational aspects of data represented as a graph. Graph-based data mining has two major approaches: frequent subgraph mining and graph-based relational learning [40]. Frequent subgraph mining and graph-based relational learning introduced briefly in this section. Then the Subdue graph-based relational learning technique will be described including substructure discovery, unsupervised hierarchical learning and supervised learning.

### 3.1.1 Frequent Subgraph Mining Approach

Graph-based data mining is the approach to finding meaningful and understandable graph-theoretic patterns in a graph which represents relational data [40]. This section depicts several approaches, mainly technologies applied to bioinformatics domain.

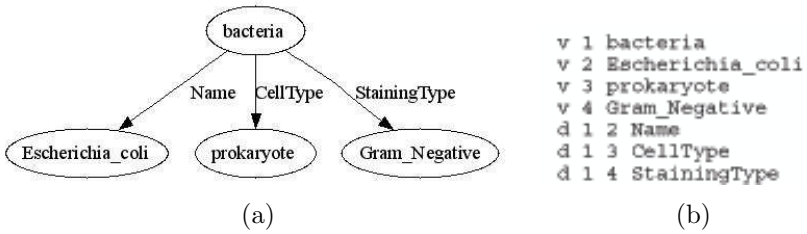
Frequent SubGraph discovery, FSG, is the approach to find all connected subgraphs that appear frequently in set of graphs represented data. FSG starts by finding all frequent single and double edge graphs. During each iteration FSG expands the size of frequent subgraphs by adding one edge to generate candidate subgraphs. Then, it evalu-

ates and prunes discovered subgraphs with user-defined constraints [41]. This approach has been applied to classifying chemical compounds [19].

Graph-based Substructure Pattern Mining, gSpan, uses the depth-first search and lexicographic ordering. First gSpan sorts the labels, removes infrequent vertices and edges and relabels the remaining vertices and edges. Next it starts to find all frequent one-edge subgraph. The labels on these edges and vertices define a code for each graph. Larger subgraphs map themselves to longer codes. If the code of $B$ is longer than $A$, the $B$ code is a child of the $A$ code in a code tree. If there are two not-unique codes in the tree, one of them is removed during the depth-first search traversal to reduce the cost of matching frequent subgraphs [42].

### 3.1.2 Graph-Based Relational Learning

Graph-Based Relational Learning (GBRL) can be distinguished from graph-based data mining in that GRBL focuses on discovery of novel, but not necessarily most frequent, substructures in a graph representation of data [43]. The main goal of GRBL is not merely to discover patterns capable of compressing the data by abstraction with instances of the patterns, but also to find conceptually important substructures to give better understanding of the data [44]. The Subdue graph-based relational learning system can perform unsupervised learning and supervised learning by substructure discovery based on Minimum Description Length (MDL). Subdue can discover patterns using background knowledge given as predefined substructures. Subdue has been applied to several areas such as Chemical Toxicity [45], Molecular Biology [8], Security [6] and Web Search [7].




(a) (b)

Figure 3.1. (a) Graph representation of *Escherichia coli* description (b) Graph file of *Escherichia coli* description.

## 3.2 Substructure discovery

Substructure discover is the technique that can mine structural data that contains not only descriptions of individual objects in a database, but also relationships between these objects. Subdue accepts input data which is represented as a graph including labeled vertices labeled directed or undirected edges between vertices. The objects and attribute values of the data are usually depicted as vertices, attributes and relationships between objects are represented as edges.

Figure 3.1 (a) shows an example of a high-level graph representation of *Escherichia Coli* (*E. Coli*). *E. Coli* which is a bacteria is categorized as Gram Negative Bacteria, which means that this bacteria is not stained dark blue or violet by Gram staining. In Figure 3.1 (b) the input file shows the syntax of the graph, where $v$ *id label* defines vertices and $d$ *id*1 *id*2 *label* defines directed edges. If $u$ is used instead of $d$, the edge is undirected.

### 3.2.1 Discovery Algorithm

Subdue's discovery algorithm is shown in Figure 3.2. The algorithm starts with three parameters: input graph, beam length and limit value. The beam length limits the length of the queue and the limit value restricts the total number of substructures considered by the algorithm. The initial state of the search is the set of substructures

```
Subdue(Graph, Beam, Limit)
    Q = {v | v is a vertex in Graph having a unique label}
    bestSub = first substructure in Q
    repeat
        newQ = {}
        foreach substructure S ∈ Q
            newSubs = Extend-Substructure (S, Graph)
                      in all possible ways
            Evaluate (newSubs)
            newQ = newQ ⋃ newSubs mod Beam
            Limit = Limit - 1
        if best substructure in new Q better than bestSub
         then bestSub = best substructure in Q
        Q = newQ
    until Q is empty or Limit ≤ 0
    return bestSub
```

Figure 3.2: Subdue's discovery algorithm

representing each uniquely labeled vertex and its instances. The *Extend-Substructure*, operator extends the instances of a substructure in all possible ways by adding a single edge and a vertex, or by adding a single edge if both vertices are already in the substructure. The substructures in the queue are ordered base on ability to compress the input graph as evaluated using the minimum description length (MDL) principle [46] which is described below. The search terminates upon reaching the limit value, or upon exhaustion of the search space.

Once the best structure is discovered, the graph can be compressed using the best substructure. The compression procedure replaces all instances of the best substructure in the input graph with a pointer, a single vertex, to the discovered best substructure. The discovery algorithm can be repeated on this compressed graph for multiple iterations as defined by the user.

### 3.2.2 Minimum Description Length Principle

The discovery algorithm of Subdue is guided by the minimum description length [47] principle. The heuristic evaluation by the MDL principle assumes that the best substructure is the one that minimizes the description length of the input graph when compressed by the substructure [46]. The description length of the substructure $S$ is represented by $DL(S)$, the description length of the input graph is $DL(G)$, and the one of the input graph after compression is $DL(G|S)$. Subdue's discovery algorithm tries to minimize $DL(S) + DL(G|S)$ which represents the description length of the graph $G$ given the substructure $S$. The compression of the graph can be calculated as

$$Compression = \frac{DL(S) + DL(G|S)}{DL(G)} \tag{3.1}$$

where description length $DL()$ is calculated as the number of bits in a minimal encoding of the graph [46].

### 3.2.3 Inexact Graph Match

Although exact substructure match can be used to find many interesting substructures, many of the most interesting patterns might show in slightly different forms. The Subdue algorithm employs the inexact graph match technique given by Bunke and Allermann [48] to allow minor differences between the instances and the substructure definition. In this inexact graph match approach, each distortion of a graph is assigned a cost. An inexact graph match is a mapping $f : V_1 \rightarrow V_2 \cup \{\lambda\}$, where $V_1$ and $V_2$ are the set of vertices of $g_1$ and $g_2$ sequently, and a graph $g_1$ is a distorted version of a graph $g_2$. A transformation $f(v) = \lambda$, where the vertex $v \in N_1$, represents a mapping from $v$ to $\lambda$. This mapping, where no vertex in $g_2$ corresponds with $v$, is called deletion. A distortion is described in terms of basic transformations such as insertion, deletion and substitution

of vertices and edge. We define the cost of an inexact graph match $cost(f)$ as the sum of the cost of each transformations using $f$. Then, a $matchcost(g_1, g_2)$ defined as the value of the least $cost(f)$ which maps $g_1$ onto $g_2$ are computed using a tree search procedure [46].

Subdue has a threshold parameter that can be specified as a value between 0 and 1, where a value of 0 allows only exact matching, and a value of 1 considers any two graphs as the same. A match threshold $t$ between 0 and 1 implies that a graph can be considered to be an instance of a substructure where $matchcost(g_1, g_2)$ is no more than $t$ time the size of $g_2$ (size$(g_2) > size(g_1)$). The size of a graph can be calculated as $size(g) = n(v) + n(e)$, where a graph $g = (v, e)$, $v$ is the vertex, $e$ is the edge and $n(x)$ is the number of $x$ [46].

### 3.2.4 Complexity

Computational complexity is the inevitable issue, because graph-based data mining usually runs with a huge amount of data like biological domain. The discovery algorithm of Subdue is computationally expensive. Subdue uses two constraints to maintain polynomial running time: Beam and Limit. Beam constraints the number of best substructures by limiting the length of newQ in Figure 3.2. Limit is a user-defined number of substructures to consider in each iteration. Inexact graph matching is the most expensive part in the Subdue algorithm. Subdue uses the branch-and bound search to guarantee an optimal solution and also limits the number of search nodes considered by each call to the inexact graph maches [46, 49].

## 3.3 Unsupervised Learning

As mentioned in the previous section, Subdue can iterate to find a new best substructure after compressing the graph with the previous substructure until the graph

cannot be compressed any more or on reaching user-defined the number of iterations. Each iteration generates a node in a hierarchical, conceptual clustering of the input data. On the $ith$ iteration, the best substructure $S_i$ is used to compress the input graph, introducing a new vertex labeled $S_i$ to the next iteration. Consequently, any subsequently-discovered subgraph $S_j$ can be defined in terms of one or more $S_i$, where $i < j$. The result is a lattice, where each cluster can be defined in terms of more than one parent subgraph.

### 3.4 Supervised Learning

The Subdue discovery algorithm has been extended to perform supervised graph-based relational learning which needs to handle negative examples. Regarding negative examples Subdue can work with two kinds of data. First, the data can be in the form of numerous small graphs, which are labeled as positive or negative examples. Second, the data can consist of two large graphs: one positive and one negative [40]. The first form is closer to the standard supervised learning problem, because we have a set of clearly defined examples. The main approach of supervised learning is to find a substructure that appears often in the positive examples, but not in the negative examples. The substructure value is increased when positive examples are covered by the substructure, but is decreased where negative examples are covered. Positive examples not covered by the substructure and negative examples covered by the substructure are considered error. The substructure value is calculated by

$$value = 1 - error \tag{3.2}$$

where the error is calculated by

$$error = \frac{\#PosEgsNotCovered + \#NegEgsCovered}{\#PosEgs + \#NegEgs} \tag{3.3}$$

#PosEgsNotCovered is the number of positive examples not covered by the substructure, and #NegEgsCovered is the number of negative examples covered by the substructure. #PosEgs is the number of positive examples remaining in the experimental set, of which the positive examples that have already been covered in a previous iteration were removed, and #NegEgs is the total number of negative examples, which is constant, because negative examples are not removed [50].

Subdue can take two approaches to minimize error. Subdue can bias the search algorithm toward a more characteristic description using the information-theoretic measure to look for a substructure that compresses the positive examples, but not negative examples. By using definition of description length Subdue tries to find and a substructure $S$ minimizing $DL(G^+ \mid S) + DL(S) + DL(G^-) - DL(G^- \mid S)$, where the last two terms represent the incorrectly compressed negative example graph. This approach will lead the discovery algorithm toward a larger substructure that characterizes the positive examples, but not the negative examples.

Instead of using the compression-based evaluation measure with error measure, Subdue can use the a set-cover approach. At each iteration Subdue adds a new substructure to the disjunctive hypothesis and removes covered positive examples. This process continues until either all positive examples are covered or no substructure exists discriminating the remain positive examples from the negative examples [51, 50]

### 3.5 Summary

This section described the Subdue graph-based data mining algorithm. Graph-based data mining is defined as finding the relational patterns in a graph representation of data. Frequent Subgraph Mining Approach and Graph-Based Relational Learning are introduced as two approaches of graph-based data mining in the first section. Subdue's discovery algorithm is described along with minimum description length (MDL) and inexact graph match. Then the supervised learning and unsupervised learning algorithms of Subdue are also introduced. Next chapter will introduce the biological network and related concepts as the domain of the Subdue application.

# CHAPTER 4

# BIOLOGICAL NETWORKS and KEGG

## 4.1 Biochemical Concepts

A biological organism has one ultimate goal: to continue the life of its species and itself in the world. This goal requires two important activities. The first is to maintain low entropy in the environment. The second is to reproduce [52, 53, 54]. To maintain low entropy means that free energy needs to be conserved at a high level. Biological organisms need to process digestion, perception, circulation, respiration, excretion and acting to maximize free energy and minimize entropy.

Every biological organism consists of one or more cells. A cell is a functional and structural basic unit of biological organisms. A cell is divided into two categories: prokaryotic cell and eukaryotic cell. A prokaryotic cell, relatively small size, does not have a nucleus. A eukaryotic cell, relatively large and full-functioned cell, has a nucleus and most cellular organelles depend on an animal cell or a plant cell. Bacteria like *E. Coli* and Salmonella germs are prokaryote organisms. Yeasts, plants and animals are eukaryote organisms [52, 53, 54]. A cell carries most of the processes for maintaining its life, because it is a functional and structural basic bio-organism by itself. The cell generates some energy from nutrients to maintain its life and reproduce. Also, it needs to protect itself against the outer environment and excrete garbages. Every activity of a cell is carried out as metabolism.

*Metabolism* is a series of enzyme-catalyzed reactions that constitute metabolic pathways in a cell or organism [52]. Each consecutive reaction in a metabolic pathway makes a specific biochemical change. Metabolism can be divided into two categories: Catabolism

and Anabolism. Catabolism is the degradation phase in which organic nutrient molecules are converted into smaller and simpler compounds, sometimes while releasing energy for biological activity. Anabolism is called biosynthesis in which small and simple molecules are built up into larger and more complex molecules, such as polypeptide, polysaccharide and triacyglycerols [52, 55, 54].

### 4.1.1 Biochemical compounds

A cell and its organelles are composed of many biochemical compounds. Biochemical compounds are a biochemical substance formed from two or more elements. For example, $H_2O$, $O_2$, $N_2$, and $CO_2$ are examples of compounds. Some of these compounds play a biochemically important role in a cell such as amino acid, glucose, lipid acid and nucleic acid. These basic compounds constitute macro molecules, much larger compounds, such as proteins, carbohydrates, lipids and genetic materials (DNA and RNA) which are working as structural, energy-resource and functional molecules [52, 55, 54].

### 4.1.2 Biochemical reaction

A biochemical Reaction is a chemical reaction which takes place in all living organisms. A chemical reaction is a process involving one, two or more compounds, which are divided into substrates and products. A substrate is a chemical compound present before a reaction, and a product is generated after a reaction. A reaction changes a substrate into a product by a chemical change or transferring some chemical groups or electron from a substrate to a product. Generally, a biochemical reaction is represented from left (a substrate) to right (a product) like the next equation.

$$6CO_2 + 12H_2O + \triangle E \rightarrow C_6H_12O_6 + 6O_2 + 6H_2(Photosynthesis) \qquad (4.1)$$

If a reaction can proceed either way ($\rightleftharpoons$), the reaction is called a reversible reaction, otherwise, an irreversible reaction [52, 55, 54]. In Figure 4.1 which describes the artificial metabolic pathway we can see several biochemical reactions. A rectangle represents a compound like a substrate and a product. In the left part of the Figure 4.1 Compound A $\rightleftharpoons$ Compound B represents the reversible reaction between compound A and B. Below this reaction, Compound C $\rightharpoonup$ Compound D represents the irreversible reaction between Compound C and D.

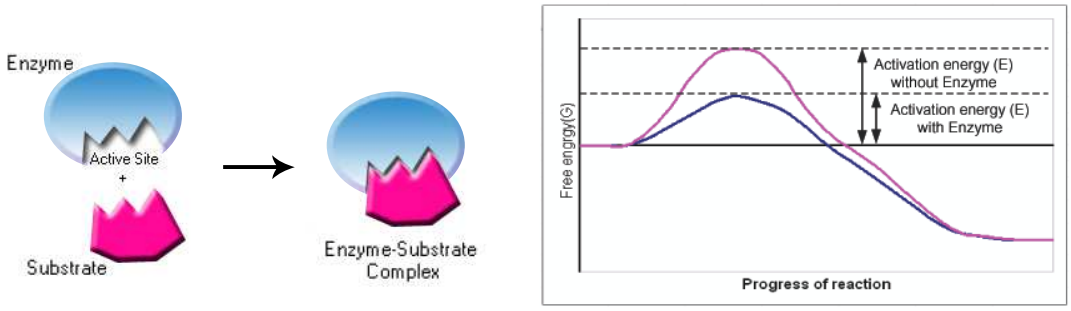
### 4.1.3 Enzyme

A catalyst accelerates the chemical reaction by providing lower activation energy between the reactants (substrate) and the reaction products. An enzyme is a powerful and specific catalyst in almost all every biochemical reaction. Except a few catalytic RNAs, almost all known enzymes are proteins. This is the main reason why a protein plays a central role in a cell or organism. A simple enzymatic reaction might be written

$$E + S \rightleftharpoons ES \rightleftharpoons EP \rightleftharpoons E + S \tag{4.2}$$

where E, S and P represent the enzyme, substrate and product; ES and EP are intermediate complexes of the enzyme with the substrate and with the product. Figure 4.2 shows that the reaction between Compound C and Compound D catalyzed by enzyme CD represented by an eclipse.

In enzyme reactions, we should understand two concepts. First, one enzyme is specific to one substrate, because the active site, which is the binding region of the substrate, has a unique geometric shape that is complementary to the geometric shape of a substrate. This is called "Lock(*enzyme*) and Key(*substrate*) Theory" in Figure 4.1 (a), because a key is used into only well-fitted lock [52, 54]. Second, an enzyme can




(a) (b)

Figure 4.1. (a) Lock and Key Theory of Enzyme-substrate complex (b) Activation energy comparing enzyme-catalyzed and uncatalyzed reaction .

catalyze a reaction by decreasing activation energy . Activation energy is the difference between the energy levels of the ground state and the transition state. The rate (speed) of a reaction reflects this activation energy. A reaction can proceed without an enzyme, but it is too slow. Lower activation energy enhances reaction rates as shown in Figure 4.1 (b) [52, 55, 54].

### 4.1.4 Pathway

A pathway is a sequence of several biochemical reactions to transform a set of substrates into a set of products [55]. From the view of computer science a pathway is a network of biochemical reactions. It is similar definition to the Internet that is the network of computer networks. It is necessary to process a series of biochemical reactions to produce macro molecules, but not just a couple of reactions, because these molecules which play central roles in the cell usually have large molecular weight and complex structures. They are produced by a cooperation of various biochemical reactions and compounds. There are always interactions among pathways. A product of a pathway may be a substrate of another pathway [52, 55, 54]. Figure 4.2 shows two artificial

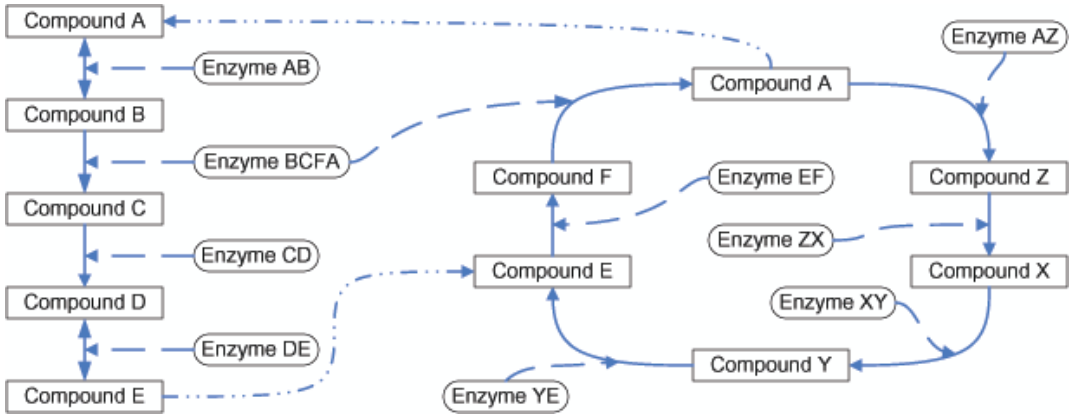



Figure 4.2. Artificial metabolic pathway. This metabolic pathway consists of two metabolic patway: Left part and Right Part.

metabolic networks (a left serial form and a right cyclic form). The Compound E is a product of the left pathway is used as a substrate in the right pathway.

### 4.1.5 Regulation

A biological organism is the most efficient system on earth. A bio-organism always chooses the most effective way to utilize its resources. For example, if a sugar level in blood is maintained as the average, the pancreas does not increase or decrease the production of insulin. If the sugar level comes down, more insulins are released from the pancreas, and if the sugar level becomes higher than the average, the production of insulin is regulated for a harmonizing concentration of sugar. An organism has a variety of mechanisms to keep its balance such as feedback control, genetic control, competitive control and allosteric control [52, 55, 54]. Moreover, all kinds of controls work in harmony with other control methods to most efficiently using its resources.

## 4.2 Systems Biology

Systems biology is an emergent field which aims at system-level understanding of biological systems. System-level understanding is an abstract concept in itself [34, 56]. System-level understanding is not only to investigate bio-systems in detail, but also to comprehend how they are working in a certain environment or situation. Finally we should be able to design the bio-systems to work optimally in natural systems. With the progress of genome sequence projects and range of other molecular biology projects that accumulate a huge amount of knowledge of the molecular nature of biological systems, we are now at the stage to seriously look into system-level understanding grounded on molecular-level understanding [57, 34]. System-level understanding, the approach advocated in systems biology, requires a shift in focus from understanding genes and proteins to understanding a system's structure and dynamics of celluar and organismal function [57, 12, 34].

There are four aspects to study: [57]

*System Structure*: First of all, we need to understand the structures of bio-systems. The primary bio-systems to be understood are the biological networks, such as the metabolism network, regulatory network and protein-protein interaction. Also we need to identify the physical structures of organisms and the mechanisms between intracellular and multicellular systems.

*System Dynamics*: After bio-systems are identified, their dynamics or behavior, also need to be understood. Fundamentally, we need to know how a system behaves over time under various conditions through metabolic analysis, sensitivity analysis, and dynamic analysis.

*The Control method*: For the purpose of application of the insights of system structures and dynamics, we need to establish a method to control the bio-systems. To maintain a bio-system in a effective life, the system is controlled optimally under

various environments. The control mechanism needs to be fully described for system-level understanding.

*The Design Method*: Ultimately, we would like to design and construct a bio-system with the aim of simulating a real bio-system. It allows us to cure severe diseases and produce a great harvest in medical science and agriculture. There are many potential applications of a bio-systems, such as simulation of disease risk, drug design, organ cloning and so on [34].

Systems biology is a new and emerging field in biology that aims at system-level understanding of biological systems. System-level understanding requires a range of new analysis techniques, measurement techniques, experimental methods, software tools, and new concepts for looking at biological systems [34].

## 4.3 Overview of Biological Networks

For system-level understanding, computational modeling of biological networks is a central research field. A biological network represents objects such as genes, proteins and other biochemical compounds, and relationships between those objects. To model a biological network allows us to fully understand not only those objects and relationships but also the dynamics of the network [58]. Nowadays, we can finally focus on computational modeling of biological networks, since we have enough results of genomics and proteomics by the means of development of new high-throughput technologies such as microarray, mass spectrometry and 2D protein gel electrophoresis [31, 38].

There are three kinds of network to process modeling: Metabolic network, protein network and genetic network. *Metabolic Network* represents the enzymatic processes within a cell, which provide energy and create parts of the cell [59, 55]. *Protein network* is the network of signal transduction networks or communication between proteins. These protein-protein interactions are mainly involved in signal pathway [55, 60]. *Ge-*

*netic network* is a regulatory network which refers to the functional inference of direct causal gene interaction. By the following, $DNA \rightarrow RNA \rightarrow Protein \rightarrow function$, gene expression is regulated at many molecular levels [61, 55]. Three categories of biological networks would be considered as two groups: Metabolic network and Protein-protein Interaction. The first one is the network of various objects and their relations, but the second one is the network of interactions between proteins [62].

## 4.4 Computational Analysis of Biological Networks

As mentioned in the previous section, the biochemical pathway is the complex interaction between molecules such as biochemical compounds, protein and other genetic materials. Efficient representation is needed for computerized analysis of biological networks. As described in the previous chapter 3 the graph-based representation would be a good choice to describe objects and relationships of the biological network. Each object like a protein, compound and gene would be a vertex and the relationships between objects would be edges. The graph representation of biological domains is a popular method. One of the most used RDBMS, Oracle, employs a Network Data Model (NDM) which enables users to model and analyze data as a graph [5].

Detecting Frequent Subgraph can find quickly frequent patterns in biological networks from KEGG PATHWAY database [63]. But the graph representation of this approach misses some features of KEGG biological networks. Mining coherent dense subgraphs technology shows better performance than frequent subgraph mining in this domain [64]. This approach compresses a group of graphs into two meta-graphs using correlated occurrences of edges for efficiently clustering. However this approach is just focused on interaction between proteins and gene products from microarray analysis.

Other approaches using the graph representation are also available. The large-scale organization is used as a framework for modeling the biological networks based on the

idea that the metabolic network has the same topological scaling properties and shows striking similarities to the inherent organization of complex non-biological networks [65]. Analysis of pattern in biological networks using the topology of the network and the directionality of its link is also tried based on large-scale topology [66]. PATHBLAST is the BLAST algorithm for biological networks that identifies conserved network regions [67].

Instead of graph theory, other approaches are used to analyze biological networks. A graphical notation and a process diagram are used to represent the pathway network [68, 69]. One approach using logic circuits tries to explain the transcriptional regulation network [70]. Mathematical expression distance is used to describe the dynamics of a cellular metabolic network [71].

### 4.5 Database of Biological Networks

Many computational technologies have given huge contributions to bioinformatics research. Like other bioinformatics areas, biological networks also have a huge amount of data. Especially, biological networks include not only many objects such as biochemical compounds, proteins and genetic materials, but also relationships between those objects. Therefore the database of biological networks has many cross-references with other databases, which have the information of chemical compounds, proteins and genetic materials. There are several databases of biological networks [62, 72, 73, 74] in the web.

Unfortunately, some of databases in Table 4.1 do not have enough information. They are still progressing to update data. Moreover they have few suitable formats to represent and distribute their data. However, a couple of database have enough data and proper format to research of biological networks. As the survey of this research, KEGG and PathCase has relatively enough data. The PathCase Pathway Database

System has 37 metabolic networks and 876 biological processes with related molecules. They PathCase has visual browser to explore pathways with related molecules [74]. It is difficult to analyze this data using user-specific methods, because their distributed format is used on the only their own browser. The KEGG PATHWAY database has not only sufficient and comprehensive data, but also a proper data format, KGML, based on XML [62].

Table 4.1. Databases of Biological Networks

| Database | URL |
|---|---|
| KEGG | http://www.genome.jp/kegg/pathway.html |
| PathCase | http://nashua.cwru.edu/pathways |
| BIND | http://bind.ca |
| DIP | http://dip.doe-mbi.ucla.edu |
| BioCyc | http://www.biocyc.org |

## 4.6 KEGG and KGML

### 4.6.1 KEGG - Kyoto Encyclopedia of Genes and Genomes-

KEGG, Kyoto Encyclopedia of Genes and Genomes, from the Kanehisa Laboratory of Kyoto University Bioinformatics Center, is a database to understand systematic function of the cell or the organism from its genomic information. KEGG has a hierarchical structure of several bioinformatics databases. KEGG has four databases at the first level, such as KEGG PATHWAY, GENES, LIGAND and BRITE [62, 2].

KEGG PATHWAY is a central database which has the information of five categories of biological networks. KEGG GENES is a collection of gene catalogs for complete genomes and some partial genomes. It consists of GENES (high-quality genomes), DGENES (Draft genomes), EGENES (EST consensus contigs), VGENES (complete viral

genomes), and OGENES (complete mitochondrial genomes, plastid ge nomes and nucleomorph genomes). KEGG LIGAND is a composite database consisting of COMPOUND (chemical compounds), GLYCAN (carbohydrate structure), DRUG (drug data), REACTION (chemical reactions), RPAIR (reactant pairs) and EZNYME (enzyme nomenclatur). Finally, BRITE is a collection of binary relations and hierarchies, which consisting of KO (KEGG Orthology: Pathway-based classification of orthologs) and cross-references to other databases [62, 2].

Biological networks, the object of this research, belong to the KEGG PATHWAY database. The KEGG PATHWAY database is not completed, but still updated. The KEGG PATHWAY database has 271 species and 167 reference networks (on August 2005). The KEGG PATHWAY database has two types of biological networks: a reference network and an organism-specific network. A reference network is a standard network which is manually generated by biologists and biochemists based on long-time accumulated experimental results. An organism-specific network is automatically generated by specific gene (coloring at Figure 4.3) in given organisms. Therefore, each organism has various numbers of biological networks, based on which genes or genetic product are found in current research [62, 2].

Of those databases, this research focuses on the KEGG PATHWAY which has information of biological networks. KEGG PATHWAY has three ways to distribute its data, graphic files, KEGG API and KGML. A first way of distribution is a graphic file map which shows the picture of a biological network in the Figure 4.3. The map is GIF format file which is generated by KEGG technician. This way is the easiest way to explain the biological network. KEGG API is the way to access the KEGG database by using SOAP technology over the HTTP protocol. The SOAP server also comes with the WSDL (Web Services Description Language), which is a XML format for describing network services, makes it easy to build a client library for a specific computer language.

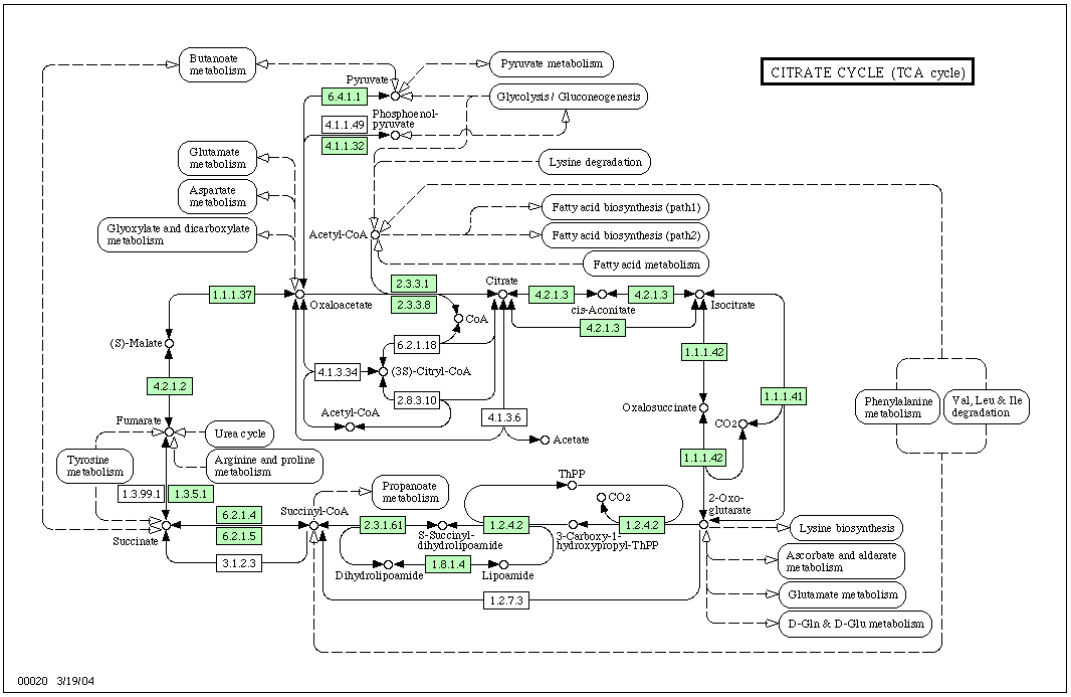



Figure 4.3. A graphic file map of TCA cycle biological network of *Homo Sapiens* [2].

This system allows users to make their own code to access the KEGG database [62, 2]. Last method is KGML which is the standard data format to express a biological network. KGML will be discussed in the next subsection [3].

As the last part of the KEGG database section, I will depict the naming convention used in the KEGG database. They usually use their convention name instead of biological nomenclature. The names of KEGG biological networks consist of several alphabet characters and several digits numbers. First, the KEGG PATHWAY DB has 271 species and one reference species which is not biologically species but same level as species. Their name is composed of three characters abbreviation of species name. For example, reference map is "map", *Homo Sapiens* (Human being) is "hsa" and *Escherichia coli*, one of the bacteria, is "eco". Map, reference network has 167 biological networks and other species has various number of networks based on research results of each species. But each network in different species is basically same even though it has a few species-specific

feature. Glycolysis in hsa is the metabolite to degrade sugar for energy generation, and Glycolysis in eco is the almost same function, wherever this metabolite is identified and stored in KEGG database. Each network is named using five digits number. For example 00010 means Glycolysis and 00251 represents Glutamate metabolism. By this manner Glycolysis in Human being is named as hsa00010 and Glutamate in *E. Coli* has eco00251 as its name. When the biological network is referred from other network, it is shown as with prefix "path" such as path:eco00010 and path:hsa00251. Other component in biological network has each convention for its name. Enzyme has a name composed of "ec:" indicating enzyme as prefix and enzyme name like $a.b.c.d$ such as ec:1.1.3.5 (Hexose oxidase) and ec:5.4.2.2 (Gulcose phosphomutase). Compound name consists of a prefix, "cpd:" and its name composed of "c" plus five digits number such as cpd:c00001 (water, $H_2O$) and cpd:c00293 (glucose, $C_6H_12O_6$). Other data in the KEGG database has each own convention like its identification number and they have cross link to each other using the convention name.

### 4.6.2 KGML - KEGG Markup Language-

The KEGG Markup Language (KGML) is an exchange format of the KEGG graph objects, based on XML. KGML enables automatic drawing of KEGG pathways and provides facilities for computational analysis and modeling of protein networks and chemical networks [3].

KGML is written in XML. XML contains the root element, which can contain other elements. And theses elements can contain child elelemts and so on. Each element can contain attributes for explain properties of the element [75]. Figure 4.3 shows some parts of eco00020 KGML file which has pathway as root and three child elements such as entry, relation and reaction.

```
<pathway name="path:eco00020" org="eco" number="00020"
title="Citrate cycle (TCA cycle)"
image="http://www.genome.jp/kegg/pathway/eco/eco00020.gif"
link="http://www.genome.jp/dbget-bin/show_pathway?eco00020">
  ...
  <entry id="15" name="eco:b0615" type="gene" reaction="rn:R01323"
  link="http://www.genome.jp/dbget-bin/www_bget?eco+b0615">
    <graphics name="citF, ybdV" fgcolor="#000000" bgcolor="#BFFFBF"
    type="rectangle" x="411" y="393" width="45" height="17"/>
  </entry>
  ...
  <relation entry1="13" entry2="14" type="ECrel">
    <subtype name="compound" value="61">
  </relation>
  ...
  <reaction name="rn:R00341" type="reversible">
    <substrate name="cpd:C00036">
    <product name="cpd:C00074">
  </reaction>
  ...
    </pathway>
```

Figure 4.4: A example of KGML [2]

As shown in the Figure 4.4, KGML has the pathway element as a root element. **Pathway** element has six attributes and three child elements. Six attributes are *name*, *org*, *number*, *title*, *image*, and *link*. *Name* is the convention name of biological network such as eco00010, hsa00020 and map00251. As mentioned above this convention name is starting with "path:". *Org* is the species name such as hsa, eco and map. *Number* specifies a five-digit pathway biological network number such as 00010, 00020 and 00251. *Title* specifies the title of this map. *Image* is the location of the graphic file of pathway map. *Link* has the resource location of the information about this pathway map in the KEGG web service [3].

The Pathway element has data has three major child elements: Entry, Relation, and Reaction.

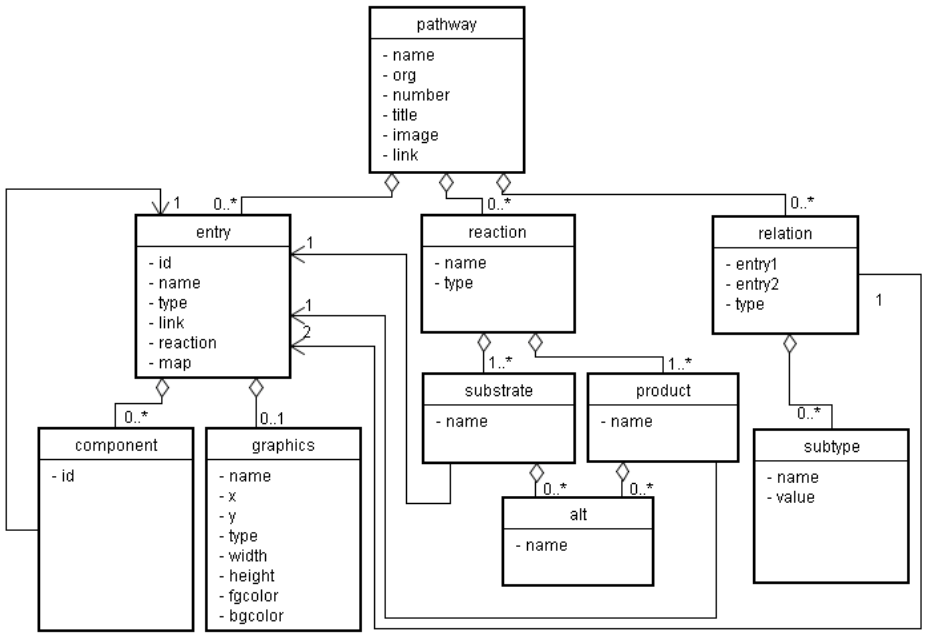



Figure 4.5. An overview of KGML. The pathway element is a root element, and one pathway element is specified for one pathway map in KGML. The entry, relation, and reaction elements specify the graph information, and additional elements are used to specify more detailed information about nodes and edges of the graph [3].

**Entry** element represent a object in the biological network such as enzyme, gene, compound and so on. Entry element has several attributes which explain the property of the Entry such as *id*, *name*, *type*, *link*, *reaction*, *map*. *Id* is the unique identification number only in each biological network, but not in the entire database. It is used for reference from reaction or relation. Entry name has the convention name as mentioned above. This name can be used for reaching other linked database such as Compound, Gene, Enzyme, and so on. *Type* indicates the type of Entry, which can be *Enzyme*, *Compound*, *Gene*, *Genes Group*, and *Map* (a name of other biological network). *Link* is the the resource location of the information about this entry. *Reaction* is the convention name of the reaction catalyzed by this entry, absolutely included in the same biological network. If this entry does not have any relationship with any reaction, this attribute is

Table 4.2. Subtype node value and Subvalue node value of the Relation element [3]

| *Subtype* value | *Subvalue* value | **ECrel** | **PPrel** | **GErel** |
|---|---|---|---|---|
| compound | *link to Entry for compound* | * | * | |
| hidden compound | *link to Entry for hidden compound* | * | | |
| activation | $--> $ | | * | |
| inhibition | $--\vert$ | | * | |
| expression | $--> $ | | | * |
| repression | $--\vert$ | | | * |
| indirect effect | $..> $ | | * | * |
| state change | ... | | * | |
| binding/association | $---$ | | * | |
| dissociation | $-+-$ | | * | |
| phosphorylation | $+p$ | | * | |
| dephosphorylation | $-p$ | | * | |
| glycosylation | $+g$ | | * | |
| ubiquitination | $+u$ | | * | |
| methylation | $+m$ | | * | |

not available. *Map* which is the Id of the map entry is specified if this entry appears in another pathway map [3].

**Relation** is relationship between protein, gene, compound and map. Relation node is a central node of relation part. Each Relation node has several attributes. First, two attributes explain basic properties of the relation such as *Type* and *Subtype*. *Type* attribute may have ECrel, PPrel, GErel, PCrel and maplink as an attribute value. *Subtype* attribute may have several values as mentioned in Table 4.2. The Subtype value may have link or its own value to give additional information of the relation. At Table 4.2. first two rows of the Subtype have a link to another Entry. Other values of *Subtype* are specific information dependent on the *Type* value of the relation. Second, the relation has two or more Entry elements (protein, gene, compound or map) as its child elements. Relation entry explains the relationships between these objects by using type and subtype [3].

**Reaction** represents a chemical reaction between one or more substrate and one or more product catalyzed by one or more enzyme. Each Reaction has two or more main entries like a *substrate* and a *product*. The other two attributes, *type* and *name*, describe properties of the reaction. By a semantics of biochemistry, the enzyme is not included as an attribute to Reaction entry. The enzyme entry has a pointer to Reaction entry to catalyze as explained above [3].

## 4.7 Summary

This chapter described the domain of this research. Some important biochemical concepts are explained for background of this research. Systems biology are introduced for post-genomic bioinformatics research. As one of the approaches of Systems biology, the study of biological networks is introduced with some examples of computational analysis. Biological network databases are represented as the result of the traditional bioinformatics research and the resource of the future work. Finally, the KEGG database which is the most comprehensive archive of the biological network is introduced as the resource of this research with its own distributed format, KGML. In the next chapter, we will describe the graph representation of the KEGG biological network and the application of the graph-based data mining as the main approach of this research.

# CHAPTER 5

# GRAPH-BASED DATA MINING FOR KEGG BIOLOGICAL NETWORKS

The main goal of this research is to apply the Subdue, graph-based data mining system, to biological networks data. Specially the goal of application is to identify how useful Subdue can be used in the research of the biological networks.

Figure 5.1 shows a flowchart of this research. At the pre-processing phase KGML data which is categorized by species was downloaded from KEGG ftp site. Every attribute is converted to graph data as a node, and each relationship is converted as edge. Because some names of entries posed potential problems when we run Subdue, this research constructs two graph representations of a biological network: the named-graph (at the Figure 5.2) and the unnamed-graph (at the Figure 5.3). The former has every unique name from KGML data, and the latter does not have any unique names. In the second representation each entry is described just by type, such as an enzyme, compound, reaction, relation and so on, but not the unique name.

In the Graph-based data mining (GDM) phase 1 Subdue runs to find the patterns in the unnamed-graph data. In this phase we run two kinds of experiments: supervised learning and unsupervised learning. Supervised learning experiments focus on distinguishing two groups of biological networks. Unsupervised learning experiments try to identify common substructure of groups of biological networks.

In the GDM phase 2 Subdue finds the patterns with the results of phase 1 as predefined substructures. The goal of phase 2 is to find complete instances of the patterns

from phase 1, which will have the unique names. The final result of this phase will be used to explore biochemical meaning of patterns.

The experiments were conducted on Intel Pentium Xeon Dual Processor 2.8GHz system running Linux kernel version 2.6.11.4 with 2GB memory.

In this chapter we will describe our approach to biological networks. The first section will introduce our graph representation to depict KGML data. The second section will explain the application of the supervised learning of the Subdue to biological networks. The last section will apply the unsupervised learning of the Subdue to biological network.

## 5.1 Graph Representation

As mentioned above a graph representation is widely used for the biological network as well as other biological domains. This section describes a graph representation for the KGML data, an example of which is shown in Figure 5.2 and 5.3. We use a directed graph, an ordered pair $G = (V, E)$, where $V$ is a set of vertices and $E$ is a set of directed edges. A directed edge $e = (\alpha, \beta)$ is considered to be directed from vertex $\alpha$ to vertex $\beta$, where $\beta$ is called the head and $\alpha$ is called the tail of the edge. Three main elements and values of all attributes are represented as vertices, and attributes and relationships between these vertices are shown as edges.

First, we will describe a detail representation with a named-graph. Then the algorithm of the conversion program is provided. Finally we will depict the unique name problem and an unnamed-graph representation.

### 5.1.1 A named-graph of Representation

This section gives a description of a named-graph representation of the biological network at Figure 5.2. KGML has *Pathway* as its root element and three child elements:

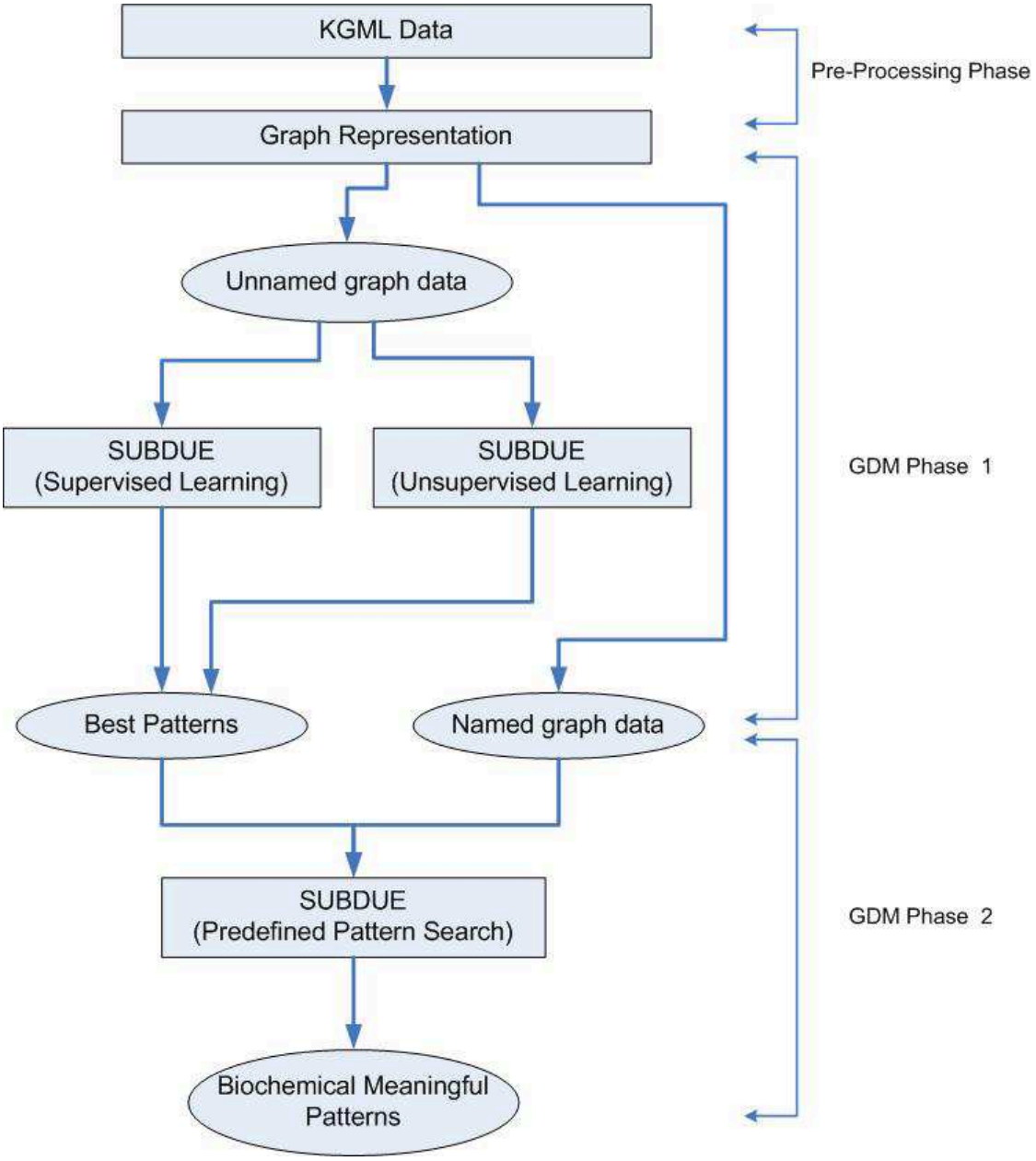



Figure 5.1. Flowchart: Application of Graph-based data mining to biological networks.

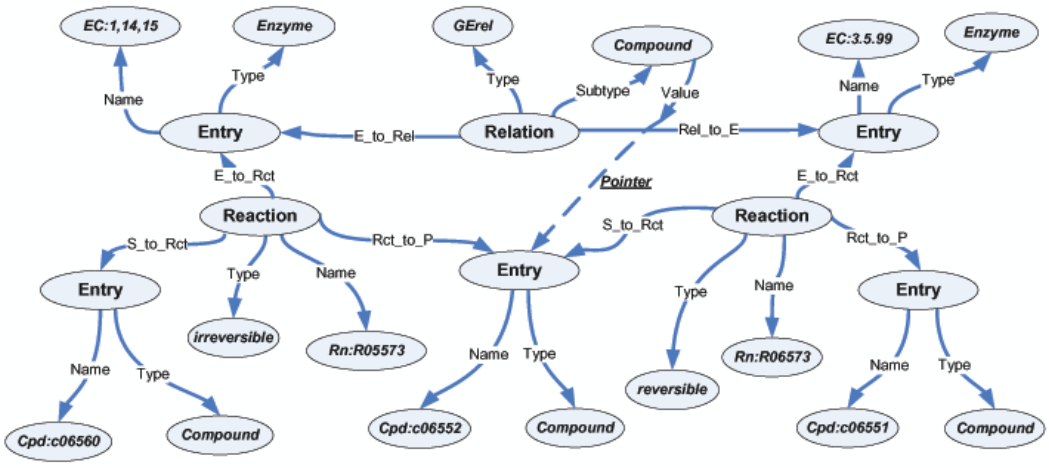



Figure 5.2. The *named-graph* representation of a biological network.

*Entry*, *Relation* and *Reaction*. As a view of representation for graph-based data mining the Pathway element is not useful to describe those elements and their relationships, because the Pathway element describes only an overview of the pathway without focusing on objects and relationships in network itself. Our approach to a graph representation starts from the three child elements.

**Entry** may have six attributes to describe the property of the entry. *Link* is not used in this research because it is a link to location of additional information. *Id* which is the unique identification number is used to represent the biological network by guidance, especially when the entry links to another entry or reaction. But it is not shown as vertex on the representation. Two attributes, *Name* and *Type* are used to mainly describe properties. These two values of Name and Type attributes represented as vertices are connected to the Entry node by *Name* attribute edge and *Type* attribute edge sequently. *Reaction* attribute is available only when the entry is the enzyme or gene which catalyzes a reaction. When the reaction attribute is available, this attribute is shown as *E_to_Rct* edge which is connected to that reaction. Every attribute is a directed edge from Entry to the value of each attribute except E_to_Rct edge. E_to_Rct

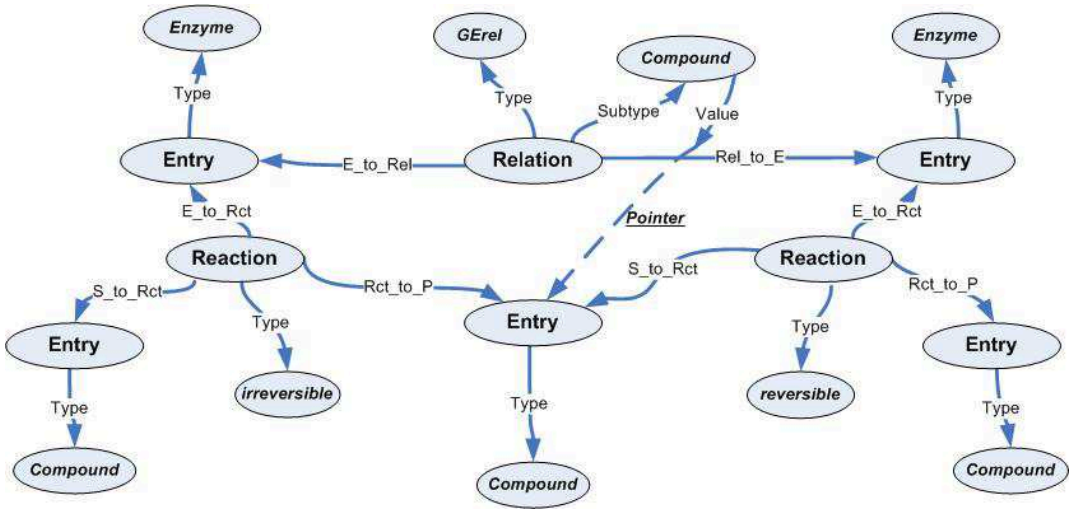



Figure 5.3. The *unnamed-graph* representation of a biological network.

edge is headed for the Entry, as a catalyst, from catalyzed the reaction. It is a general idea in biochemistry that the enzyme is considered as an assistant to the reaction even though it is almost always mandatory [52].

**Relation** shows the relationship between protein, gene, compound and map. The Relation node has several attributes. Two value vertices of *Type* and *Subtype* attributes are connected to the Relation vertex by a *Type* attribute edge and a *Subtype* attribute edge respectively. As mentioned in the last chapter, the Subtype attribute may have a subvalue or a pointer to another Entry. If Subtype has a subvalue, Subtype is represented as a *Subvalue* vertex connected to the Relation vertex by a *Subtype* edge. In another case where the Subtype has compound or hidden compound as its subvalue, the subvalue vertex is connected to the Relation vertex by the *Subtype* edge, and then the *Subtype* vertex is connected to another Entry by a *Value* edge as a pointer role. Relation element has two Entry names, which are already represented as *Entry* vertices as above, as its child elements. Usually these are same kinds of Entry, but they are ordered as first and second. First and second vertices are connected to Relation vertex by a *E_to_Rel* edge

and a *Rel_to_E* edge sequently. Every edges are directed from Relation vertex to the value of each attribute. If Value edge is available, it is headed for the pointed Entry from the Subtype vertex.

**Reaction** represents a chemical reaction between one or more substrate and one or more product catalyzed by one or more enzyme. *Reaction* vertex has two attributes: type and name. Values of two attributes as vertices are connected to Reaction vertex by *Type* and *Name* edges respectively. Reaction element has two or more child elements are categorized as a Substrate and a Product, which are already represented as Entry as above. A Substrate vertex is connected to Reaction vertex by a *S_to_Rct* edge, and as a Product vertex is connected by a *Rct_to_P* edge. As mentioned above, Reaction entry is connected to the Entry which is catalyzing this Reaction by *E_to_Rct* edge. The directions of all edges are headed for all attribute vertices and child vertices from Reaction vertex.

### 5.1.2 Converting KGML to a Graph: KGML2Graph

In this research KGML2Graph was implemented to convert KGML data to a graph using XML access technology since KGML is written by XML. To access XML we use the DOM library in the Java Development kit (Version 1.4.2). The DOM is a platform and language neutral interface which allows us to access and update the content, structure, and style of documents. The DOM model not designed for just Java, but to represent the content and model of documents across all programming language and tools [76, 77]. Using the DOM library, well-structured documents in XML can be accessed efficiently.

Figure 5.4 shows the algorithm of the conversion program KGML2Graph. The algorithm starts by tracing the tree structure of the KGML document. First it visits the root node, *Pathway*, which is not used in this representation. Then it visits each child node, such as *entry*, *relation* and *reaction*. When it visits each child node, it

gets the information of node and its attributes. Then this information is stored to the queue. KGML2Graph has three major queues for three child elements. After completing the storage of all information from the KGML, it starts to print the nodes and their attributes as the vertices of a graph. Then it prints each relationship between vertices and attributes as the edges of the graph.

```
Get Node from KGML DOM document
while Node is not null do
    if Node.getNodeName() equals "entry" then
        Node.getAttribute()
        Put Attribute data categorized by each entry into the Queue
    if Node.getNodeName() equals "relation" then
        Get Attribute data for each relation from the Queue
        Construct relation with entry
    if Node.getNodeName() equals "reaction" then
        Get Attribute data for each reaction from the Queue
        Construct reaction with entry

Print Node and Attributes from the Queue as the vertex
Print Relationships between vertices as the edges
```

Figure 5.4: KGML2Graph conversion algorithm

### 5.1.3 An unnamed-graph representation

A named-graph representation has almost all features of KGML biological networks. Some unique names of entries posed potential problems when we run Subdue on the the named-graph, because KGML data has several unique names. For example an enzyme with ec:3.1.3.9 as its name acts as D-glucose-6-phosphate phosphohydrolase in the Glycolysis biological network. Generally this enzyme is found only in the glycolysis pathway. Therefore if we try to find the best substructure to distinguish this network from others, just one pattern, this name, is found by Subdue. There is less need to have the unique name of each entry, because our research goal is to analyze patterns of biolog-

ical networks. For this reason this research constructs an unnamed-graph representation of KGML data as well as the named-graph representation.

The unnamed-graph representation is not much different from the named-graph. Three elements such as entry, relation and reaction have their own name represented as the *Name* node and the *Name* edge. These nodes and edges are erased in the unnamed-graph. By this way the unnamed-graph does not have unique label but contains other properties of all elements (Figure 5.3).

The goal of this research is to find interesting and meaningful patterns of relations between objects from biological networks. The unnamed-graphs has enough information to explore the relations. After finding the patterns, the pattern is evaluated using the named-graph representation.

## 5.2 Supervised learning

The main goal of supervised learning is to distinguish between two groups of networks. Graphs of biological networks are divided into two groups: positive examples and negative examples. Subdue searches for some patterns which are presented in one group but not in another group. In the supervised learning section, two kinds of experiments are processed. First one is to distinguish between one network in some groups of species and another network in the same group. It is a classification by kinds of biological networks. Second one is to distinguish between some groups of networks in one species and same groups in another species. It is a classification by species.

Table 5.1 and 5.2 show the experimental sets in supervised leaning. Set represents experimental sets. The name of a set consists of two parts, XP and XN, which represent the network name of the positive examples and negative examples respectively. The number of examples represents a number of positive and negative examples. Source in Table 5.1 represents source groups of species. This table has three groups: Eukaryote

Table 5.1. Experimental set used in classification by the biological network

| **Set** (XP_XN) | **Number of examples** (Pos./Neg.) | **Source** (Species group) |
|---|---|---|
| 00240_00230 | 17/17 | Eukaryote |
| 00230_00240 | 17/17 | Eukaryote |
| 00300_00310 | 9/16 | Eukaryote |
| 00310_00300 | 16/9 | Eukaryote |
| 00520_00530 | 14/17 | Eukaryote |
| 00530_00520 | 17/14 | Eukaryote |
| 00061_00010 | 15/17 | Eukaryote |
| 00010_00900 | 44/41 | 45 Set |
| 00240_00230 | 45/45 | 45 Set |
| 00251_00010 | 45/44 | 45 Set |
| 00010_00510 | 31/44 | 45 Set |
| 00010_00230 | 44/45 | 45 Set |
| 00061_00010 | 44/41 | 45 Set |
| 00010_00900 | 149/143 | 150 Set |
| 00061_00100 | 140/149 | 150 Set |

set, 45 Set and 150 Set. Each set has a different numbers of species. Eukaryote set consists of all eukaryote species (17) in the KEGG PATHWAY database. 45 Set has 45 species and 150 Set has 150 species, which are from each species group. The number of positive and negative examples is less than or equal to the number of each source group, since the example network may not yet be constructed (or not presented) in the specific species. For example all 17 species of the eukaryote cell have the 00010 network. But, *Encephalitozoon cuniculi* (fungi) and *Danio rerio* (Zebra fish) do not have the 00061 network.

### 5.2.1 Classification by the biological network

First classification tries to find the patterns that are able to distinguish between two groups of biological networks. In each experimental set we have one biological network

Table 5.2. Experimental set used in classification by species

| **Set** (XP_XN) | **Number of examples** (Pos./Neg.) |
|---|---|
| hsa_eco | 139/103 |
| hsa_eco:25 | 25/24 |
| eco_sty | 120/120 |
| eco_bsu | 103/97 |
| sce_eco:25 | 24/24 |
| mmu_sce | 90/150 |

in the one species group and another biological network in the same species groups. For example 00061_00010:Eukaryote set has two groups: The positive example (XP) group has the 00061 biological network in the Eukaryote species group. The negative example (XN) group has the 00010 biological network in the same group. The table 5.1 shows every set used in this experiment. The first six consist of pairs to identify difference of accuracy when positive examples and negatives examples are exchanged to each other.

The experiment is processed by following the flowchart in Figure 5.1. Subdue runs with a graph file containing all positive examples and negative examples in the one set to find some substructures that are in the positive examples, but not in the negative examples. This phase is called GDM phase 1. After finding patterns to distinguish between the positive and negative examples, we run Subdue on the named graph data with the result of phase 1 as the predefined substructure option in Subdue. It is phase 2. Phase 2 is for getting whole patterns of the original biological network. After phase 2, we are able to compare these patterns with the examples in phase 1 to find some biological meaning.

### 5.2.2 Classification by species

The second supervised learning experiment is to distinguish between a group of biological networks in one species and the same group of biological networks in another species. Two sets, the hsa_eco:25 and the sce_eco:25 use 25 biological networks sets, and other use all biological networks in the species. This experiment is a classification by species. The experiment process is the same as the first supervised learning experiment except for the difference in the experimental set.

## 5.3 Unsupervised Learning

Unsupervised learning tries to find common substructures across several groups. The ultimate purpose of applying clustering to biological networks is to gain a better understanding of the networks by using hierarchical topologies.

In the unsupervised learning two kinds of experiments are processed. First one is to find common patterns in one kind of network across a group of species. It is for finding common substructures across all species to describe the biological network. Second trial is to find a common substructures in a group of networks in one species. This experiment allows us to understand what common structures the different networks have.

### 5.3.1 Clustering in species

Table 5.3 shows the experimental sets in clustering in species. Set represents the name of experimental set. First part of the name represents the name of biological network and second part represents the source group. Four source groups are used such as eukaryote, 45, 150 and All (all species) set. The 00010_euk set is the set of 00010 networks from eukaryote species group. It has 15 examples out of 17 eukaryote species. Every species does not have every biological network as mentioned in last section.

Table 5.3. Experimental set used in clustering in species

| **Set** (Network_Src) | **Number of examples** (Positive examples) |
|---|---|
| 00010_euk | 17 |
| 00061_euk | 15 |
| 00010_45 | 44 |
| 00230_45 | 45 |
| 00251_45 | 45 |
| 00510_45 | 31 |
| 00900_45 | 41 |
| 00061_150 | 140 |
| 00010_all | 268 |
| 00061_all | 246 |

Table 5.4. Experimental set used in clustering in networks

| **Set** (Name of species) | **Number of examples** (Positive examples) |
|---|---|
| ath | 102 |
| dme | 100 |
| eco | 103 |
| rno | 120 |
| sce | 90 |
| mmu | 130 |
| hsa | 139 |

This experiments finds a common structure in one sort of network across the a group of species. After preparing some samples, GDM phase 1 is processed. Every example in a set is placed as a positive example. Subdue runs on this example with the 10 iterations option. After phase 1, we can get a hierarchical tree of patterns. In the GDM phase 2 we can recover the erased names for the most interesting pattern of phase 1 to know its biochemical meaning.

### 5.3.2 Clustering in networks

This experiments finds a common structure in every network in a species. The process follows the same way as the previous section. Table 5.4 shows the experimental set in clustering in networks. Set represents a name of species. Positive examples represents the number of biological networks in this species. For example, hsa set has 139 positive examples from human.

## 5.4 Summary

This chapter described the graph-based data mining approach to biological networks from the KEGG PATHWAY database. We explained the graph representation biological networks. We then described the supervised learning of Subdue and experimental data sets. The last section provided the unsupervised learning approach and experimental data set. Next we will give results of these experiments and discuss the biological meaning of the results.

# CHAPTER 6

# RESULTS AND DISCUSSION

This chapter shows results from two approaches, supervised learning and unsupervised learning, which are described in the previous chapter. Then the biological meaning of the substructures which are found by Subdue are investigated based on a variety of KEGG databases as mentioned in chapter 4. The motivation of this exploration is to prove that the substructure found by graph-based data mining is biologically important and meaningful. Each results also shows the accuracy (in the supervised learning) and the running time. In this research we have more focus on finding meaningful patterns than the issue of the running time.

## 6.1 Supervised Learning

Supervised learning using graph-based relational concept learning is for distinguishing positive examples from negative examples. The ultimate goal is to find novel and biologically understandable patterns to be able to classify the two groups.

The results of supervised learning experiments show quite different results between two approaches. The first approach, classification by the biological network, allows us to distinguish fairly efficiently between positive examples and negative examples. It is reasonably clear in a sense that each biological networks has quite a different structure for their functions. The second approach of distinguishing between two groups of species does not perform as well. It is assumed that biochemical pathways dose not show species-specificity, not like proteins. In this section we show the results of supervised learning approaches and discusses the results.

Table 6.1. Results of classification by the biological network

| **Set** (XP_XN:src) | **Num. of Examples** (Pos./Neg.) | **Size** (V+E) | **Accuracy** (%) | **Running Time** (sec.) |
|---|---|---|---|---|
| 00230_00240:euk | 17/17 | 75,086 | 100.00 | 166.22 |
| 00240_00230:euk | 17/17 | 75,086 | *55.88* | 1455.36 |
| 00300_00310:euk | 9/16 | 14,715 | 100.00 | 8.54 |
| 00310_00300:euk | 16/9 | 14,715 | *64.00* | 12.44 |
| 00520_00530:euk | 14/17 | 15,689 | 83.87 | 17.76 |
| 00530_00520:euk | 17/14 | 15,689 | 100.00 | 8.72 |
| 00061_00010:euk | 15/17 | 56,914 | 100.00 | 458.97 |
| 00010_00900:45 | 44/41 | 88,041 | 100.00 | 810.81 |
| 00240_00230:45 | 45/45 | 183,701 | *66.67* | 9420.11 |
| 00251_00010:45 | 45/44 | 129,187 | *60.67* | 14908.60 |
| 00510_00910:45 | 31/44 | 482,767 | 100.00 | 905.14 |
| 00010_00230:45 | 44/45 | 179,393 | *61.80* | 3679.12 |
| 00061_00010:45 | 44/41 | 117,582 | *48.19* | 12494.61 |
| 00010_00900:150 | 149/143 | 286,091 | 88.70 | 4253.33 |
| 00061_00010:150 | 140/149 | 371,032 | *48.44* | 13374.54 |

### 6.1.1 Classification by the biological network

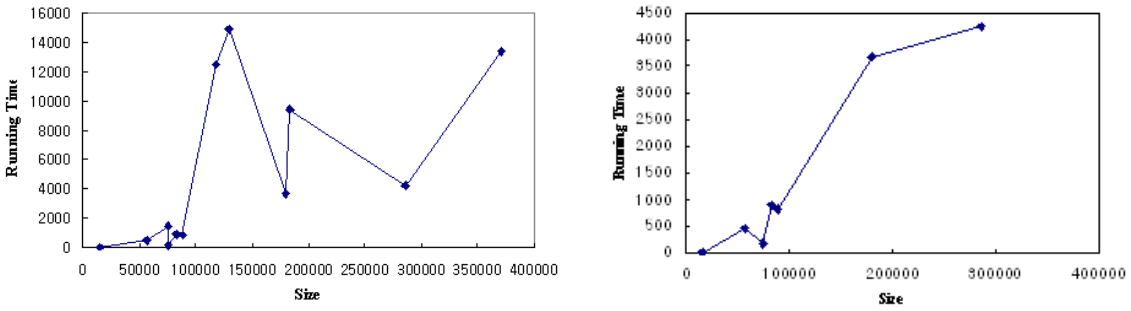
As shown in Table 5.1, several experimental sets are used to process this approach. For most cases Subdue can find a substructure to discriminate clearly between two examples.

We tried to use MDL and set-cover option as the evaluation method. In classification tests set-cover is working better at the view of performance and accuracy. The number of iterations parameter is set to the same value as the number of positive examples in an experimental set. to be sure to cover all positive examples, even though Subdue usually iterates less than those times. The Limit value is set to 50 or 100, because each experimental set has around twelve initial substructures. The limit value should be greater than the number of initial substructures. But after this restriction limit value is defined base on several trials.

Table 6.1 shows the results of the classification by the biological network. Set represents experimental sets. A name of set consists of three parts: XP, XN and src. XP and XN represent the network name of positive examples and negative examples respectively. Src represents source sets, as described in the last section. Number of examples represents a number of positive and negative examples. The size of the graph can be calculated as $size(g) = n(v) + n(e)$, where a graph $g = (v, e)$, $v$ is the vertex, $e$ is the edge and $n(x)$ is the number of $x$. The size in Table 6.1 is calculated as $size(XP) + size(XN)$. Accuracy is calculated as $(TP + TN)/(NXP + NXN)$, where $TP$ is the number of the positive examples containing at least one of the best patterns from any iteration, $TN$ is the number of the negative examples containing none of the best patterns from any iteration, XPN is the number of positive examples, and $NXN$ is the number of negative examples.

Most cases have more than 60% accuracy. Usually Subdue can find substructures to distinguish two examples clearly except for a last couple of iterations. In the 00061_00010:150 set the substructures are found in the only positive examples before last iteration. But, the substructure in the last iteration is found in the four positive examples and all negative examples.

In these 00230_00240, 00300_00310 and 00520_00530 sets, the experiment ran twice with a different way. At first trial 00230, 00300 and 00520 are set as positive examples, and at second trial they are set as negative examples. At all cases, the second trials yields 100% accuracy because these three networks, 00240, 00310 and 00530, have an *ortholog* entry, but not in the negative examples. As the view of supervised learning, this classification is successful. However this case is not the general case. Ortholog is a gene that has the same function and origin in different species. Some biological networks of which ortholog research is completed has ortholog entry. Even though Subdue can distinguish easily, it does not generate special biological meaning.




(a) (b)

Figure 6.1. Running time with graph size (a) all results, (b) sets with more than 80% accuracy in classification by species.

Running time varies by not only size of graphs but also other factors like the structure of the graphs. Running time of the 00230_00240:euk set is almost nine times faster than 00240_00230:euk sets, because 00240_00230:euk has a clear pattern, the ortholog vertex to distinguish two samples. Subdue runs in polynomial time with user-defined Limit and Beam as described in the Chapter 3. When Subdue has hard time to find substructure to distinguish clearly two examples, running takes more time. Figure 6.1 shows the running time with the graph size. It compares all results (a) with sets containing more than 80% accuracy (b). Figure 6.1 (b) shows that Subdue's running time increased polynomially if it can distinguish clearly. Figure 6.1 (a) describes that running time shows quite a different trend if Subdue cannot find the best pattern in just positive examples, not negative examples.

One experiment, 00010_00900:45, will be given to discuss as an example. This set has 100% accuracy, and it does not have any exceptional case. By discussing the result, we try to prove that the substructure which is found by the Subdue has understandable biological meaning. In this example Subdue runs four iterations, and at each iteration it found the best patterns often found in positive examples, but not in the negative examples. At the first iteration the best pattern (Figure 6.2) was found in forty instances

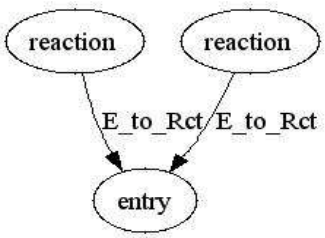



Figure 6.2. First best patterns from 00010_00900 classification.

of forty positive examples, but not in any negative example. After GDM phase 1 we can know the first best pattern means that one entry is related to two reactions. Because *E_to_Rct* is the relation between reaction and enzyme (gene), the entry should be the enzyme or the gene.

At the GDM Phase 2, Subdue run on the named-graph as of the same example of 00010_00900:45 set with the first best pattern (Figure 6.2) as the predefined substructure. Subdue can find clearly all forty instances in the named-graph, too. By using Phase 2 we can add more vertices and edges which are erased in the unnamed-graph or are not found at Phase 1. The substructure of Figure 6.3 is the updated pattern from the result of Phase 1 and is the final result of this experiment. The vertices and edges marked by "[ ]" are included from the original substructure from the GDM phase 1.

The pattern of the final result shows two reversible reactions, R01063 and R01061, which are catalyzed by one enzyme from the gene, aae:aq_1065. In fact the location of entry, aae:aq_1065 should be enzyme, not gene. However KEGG pathway shows the gene instead of the enzyme when the enzyme is made from the ortholog gene. Ortholog gene is the gene that has the same function and origin in different species. If the enzyme can catalyze a reaction $\Omega$ in the species $X$, $Y$ and $Z$, the enzyme $\alpha, \beta, \gamma$ are from the ortholog gene. KEGG Pathway tries to describe the enzyme from the ortholog gene in this way.

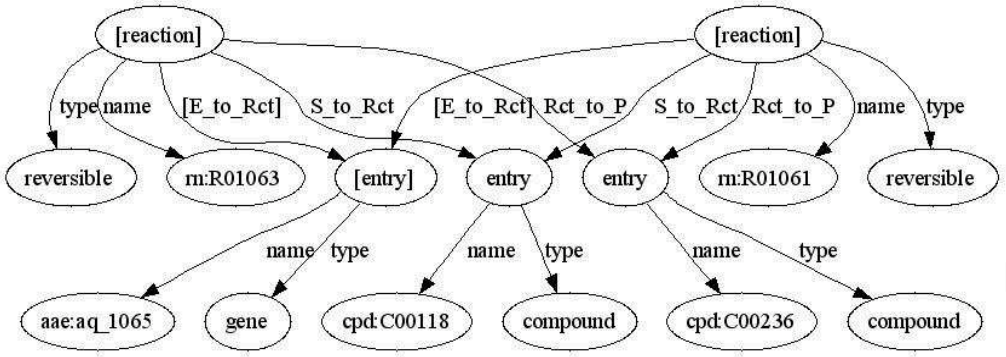



Figure 6.3. Updated substructure of first best patterns from 00010_00900 classification.

Thus gene aae:aq_1065 can be considered as enzyme. The original enzyme name having ec:1.2.1.12 as id is glyceraldehyde-3-phosphate dehydrogenase. This enzyme catalyzed two reactions R01061 (equation 6.1 and Figure 6.5) and R01063 (Equation 6.2 and Figure 6.6). These two reactions and the enzyme ec:1.2.1.12 are shown in the Figure 6.4 The official name of the 00010 biological network is *Glycolysis*.

Glycolysis is a preprocessing reaction of the energy generating reaction which degrades a molecule of glucose (6 carbon) in a series of enzyme-catalyzed reactions to yield two molecules of the three-carbon compound, pyruvate (equation 6.3). In the glycolysis biological network the most important materials are NADH, NADPH and ATP since they are energy-related compounds. When the bio-organism intakes nutrients, it digests them into primary elements such as glucose, amino acid and lipid acid. Then each cell degrades those elements to generate ATP which is energy material in the cell. Therefore the reaction regarding three energy-related compounds is one of the most important reaction biochemically. The glycolysis pathway is the starting point of the energy-generating mechanism. The pattern which is found by the Subdue is the part of NADH and NADPH related reactions which is placed in the 00010 networks, but not in 00900 network which

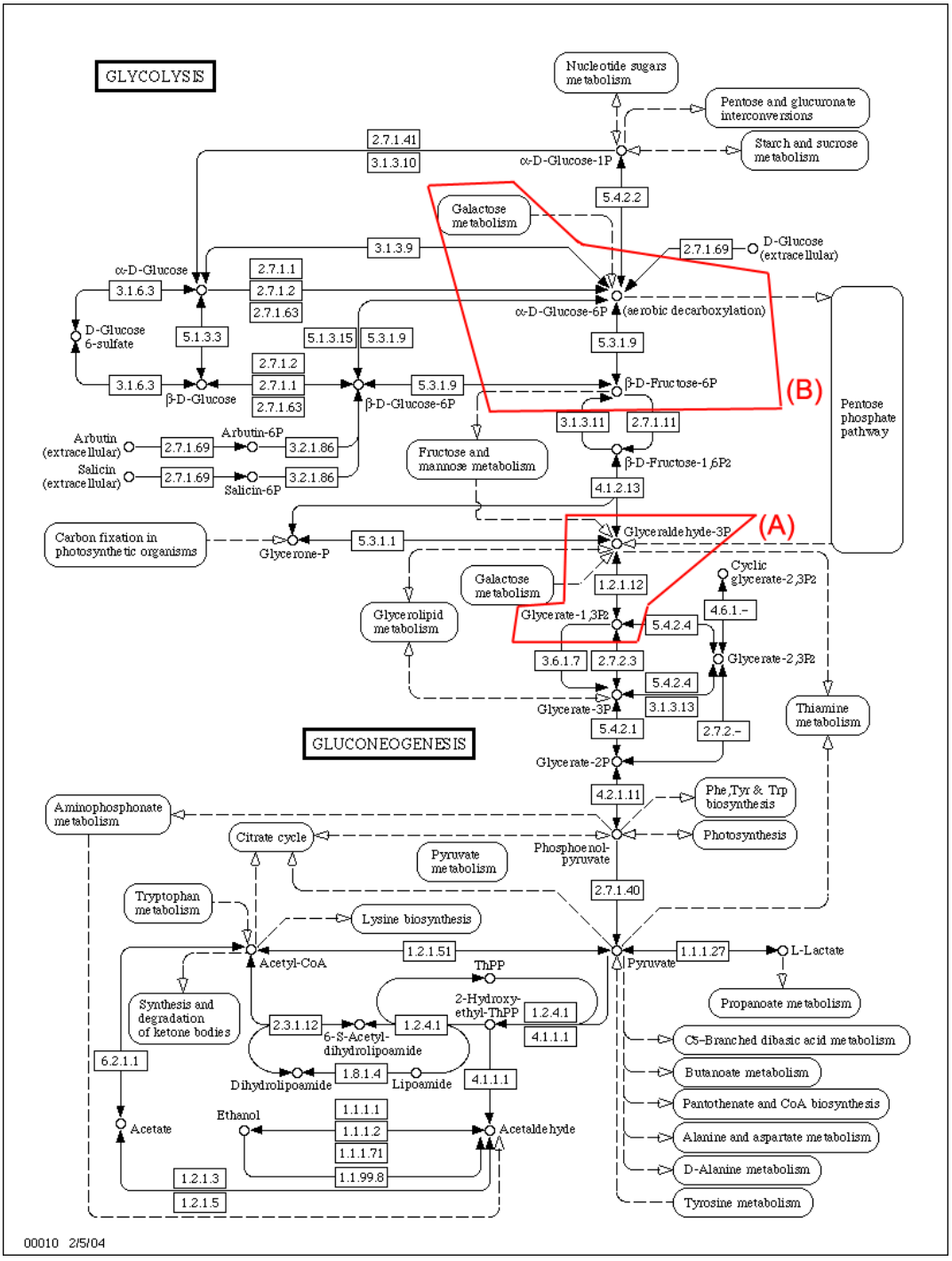



Figure 6.4. A graphic file map of Glycolysis biological network of reference network [2].

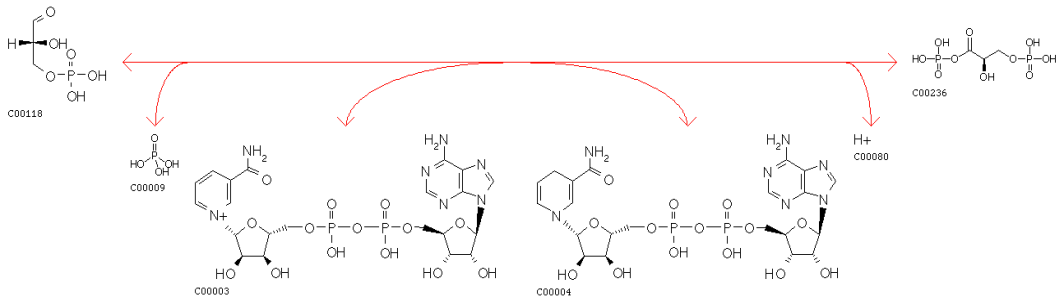



Figure 6.5. Reaction R01061 [2].

is Terpenoid biosynthesis. By the reason of this result, the substructure found by Subdue can have understandable biological meaning.

$$C_3H_7O_6P + H_3O_4P + NAD^+ \leftrightarrow C_3H_8O_{10}P_2 + NADH + H^+ \tag{6.1}$$

$C_3H_7O_6P$ : D-Glyceraldehyde 3-phosphate

$H_3O_4P$ : Orthophosphate

$NAD^+$ : $C_{21}H_{28}N_7O_14P_2$, Nicotinamide adenine dinucleotide

$C_3H_8O_10P_2$ : 3-Phospho-D-glyceroyl phosphate

$NADH : C_{21}H_{29}N_7O_{14}P_2$

$$C_3H_7O_6P + H_3O_4P + NADP^+ \leftrightarrow C_3H_8O_{10}P_2 + NADPH + H^+ \tag{6.2}$$

$NADP^+$ : $C_{21}H_{29}N_7O_{17}P_3$, Nicotinamide adenine dinucleotide phosphate

$NADPH : C_{21}H_{30}N_7O_{17}P_3$

$$C_6H_{12}O_6 + 2NAD^+ + 2ADP + 2P_i \rightarrow 2CH_3COCOO^- + 2NADH + 2H^+ + 2ATP + 2H_2O \tag{6.3}$$

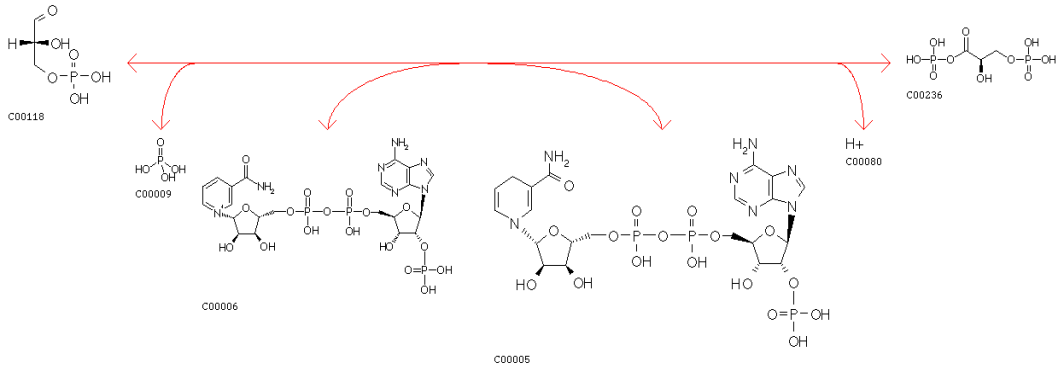



Figure 6.6. Reaction R01063 [2].

Table 6.2. Results of classification by species

| **Set** (XP_XN:src) | **Num. of Examples** (Pos./Neg.) | **Size** (V+E) | **Accuracy** (%) | **Running Time** (sec.) |
|---|---|---|---|---|
| hsa_eco | 139/103 | 190,719 | 57.85 | 5960.06 |
| hsa_eco:25 | 25/24 | 60,840 | 71.43 | 199.44 |
| eco_sty | 120/120 | 274,864 | 52.08 | 16227.67 |
| eco_bsu | 103/97 | 144,751 | 52.50 | 2441.80 |
| sce_eco:25 | 24/24 | 54,897 | 54.17 | 278.18 |
| mmu_sce | 90/150 | 164,956 | 38.33 | 13985.84 |

### 6.1.2 Classification by species

Table 6.2 shows the result of classification by species using the same way as described above. Each experimental set consists of two species as positive and negative example. In classification by species Subdue cannot find substructure to distinguish clearly between two examples. At each iteration Subdue finds the patterns which are not only in positive examples but also in negative examples. In the example of the hsa_eco set, Subdue finds the patterns which are found in both sets of examples except one negative example. Accuracy of the results are shown at Table 6.2. Accuracy may not look bad. However even though the result of classification by species has the same accuracy as some in classification by network, it does not have the same meaning. In the

approach by network most of set can distinguish between two examples before the last couple of iterations. Only last one or two iterations have the best pattern which is found in both sides. In the approach by species from first iteration to last iteration Subdue cannot distinguish clearly, and the best pattern is found in both sets of examples at most iterations.

The reason is explained as follows. Biologically species-specificity is one of most important concepts. However species-specificity does not affect the biological network of KEGG. Basically species-specificity is oriented from protein structure and gene sequence. Biological network is the system which is composed of proteins, gene products, compounds and their relationships. Bacteria have 600,000 DNA base pairs, and humans have 3 billions DNA base pairs. Also protein structure and sequences from those genes are different across all species. However the glycolysis metabolic pathway is not quite as different for each species. The difference in metabolic pathways in each species comes from contained molecular structures, not from the pathway itself [52, 54]. Moreover the KEGG biological networks of each species are generated automatically based on reference networks and several specific molecules of the species [62]. Therefore KEGG pathway does not contain species-specificity in each species.

Figure 6.7 shows the running time with graph size from Table 6.2. Even though Subdue cannot distinguish clearly two examples, it runs in roughly polynomial time in most cases. Since Subdue generates similar results as when the best patterns cannot distinguish clearly all the cases, the trend of running time is similar.

## 6.2 Unsupervised Learning

In unsupervised learning all samples are placed as positive examples in the set graph file. This experiment uses MDL to evaluate substructures. To get the hierarchical tree

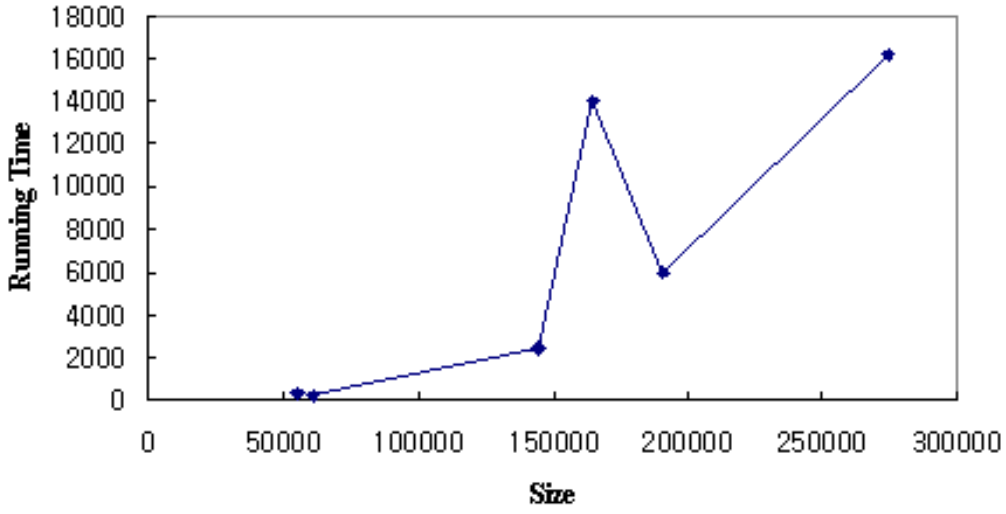



Figure 6.7. Running time with graph size in classification by species.

of substructures Subdue is iterated for twenty times. Unsupervised learning by Subdue allows us to understand better the structure of the graphs.

### 6.2.1 Clustering in species

Clustering in species is to find the common patterns in one network across a group of species. Because a biological network is a network of a variety of biochemical reactions, there are easily assumed to exist several common patterns. After processing several experiments Subdue can identify some common substructures of each group. Table 6.3 shows the result of clustering in species. Set represents a name of the network and the source set. Positive examples means the number of examples in a set. Size of each set is total size of graphs in a set. A size of the graph can be calculated as $size(g) = n(v)+n(e)$, where a graph $g = (v, e)$, $v$ is the vertex, $e$ is the edge and $n(x)$ is the number of $x$. Iteration shows the number of iterations run by Subdue. Running time shows the time to be taken to run Subdue on a set. Figure 6.8 shows the trends of Subdue's running

Table 6.3. Results of clustering in species

| **Set** (Network_Src) | **Positive examples** (Number) | **Size** (V+E) | **Iteration** (Number) | **Running Time** (sec.) |
|---|---|---|---|---|
| 00010_euk | 17 | 28,516 | 10 | 425.03 |
| 00061_euk | 15 | 28,398 | 10 | 566.23 |
| 00010_45 | 44 | 66,860 | 10 | 915.27 |
| 00230_45 | 45 | 112,893 | 10 | 3498.42 |
| 00251_45 | 45 | 68,327 | 10 | 1468.25 |
| 00510_45 | 31 | 15,907 | 10 | 32.42 |
| 00900_45 | 41 | 21,181 | 10 | 127.36 |
| 00061_150 | 140 | 147,874 | 10 | 3883.73 |
| 00010_all | 268 | 413,885 | 10 | 57418.31 |
| 00061_all | 246 | 249,331 | 10 | 13608.48 |

time of clustering in species. It shows polynomial running time. 00010_all set has the largest size of the input graph and shows the longest running time. This size is 14 times larger than 00010_euk and it takes 135 times longer than the shortest one.

Figure 6.9 (a) shows one best substructure which is found in 00061_150 (3,217 instances in 140 examples), 00061_all (5,494 instances in 246 examples) and 00230_45 (2,185 instances in 45 examples) at the first iteration, 00061_euk (436 instances in 10 examples) at the second iteration, and 00510_45 (306 instances in 7 examples) at the third iteration. Figure 6.9 (b) is the best substructure found in 00010_euk (264 instances in 17 examples), 00010_45 (740 instances in 44 examples) and 00010_all (4,609 instances in 268 examples) at the first iteration and 00900_45 (127 instances in 7 examples) at the fourth iteration. An observation of two pictures allows us to identify substructure (a) as a part of (b). As a matter of fact, substructure (a) would be a basic model of a biochemical reaction. This substructure describes an enzyme-related reaction, equation 4.2 in chapter 4. It is natural that all of biological networks have this pattern.

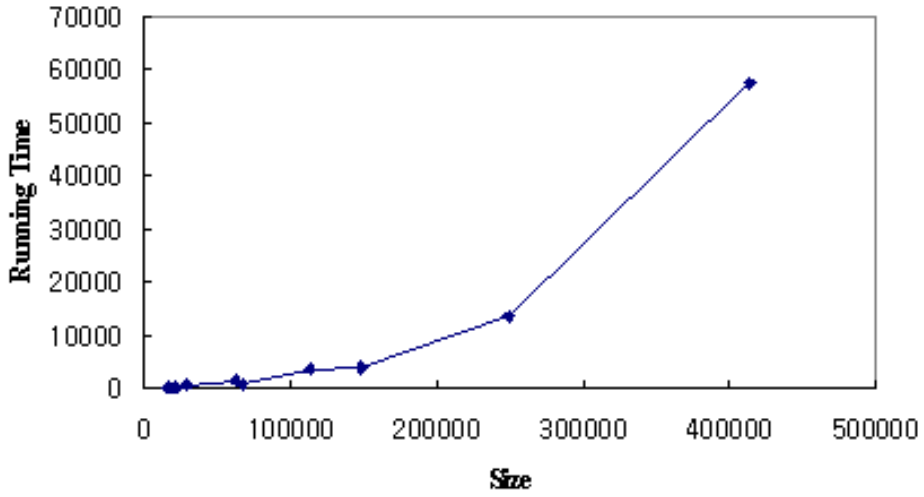
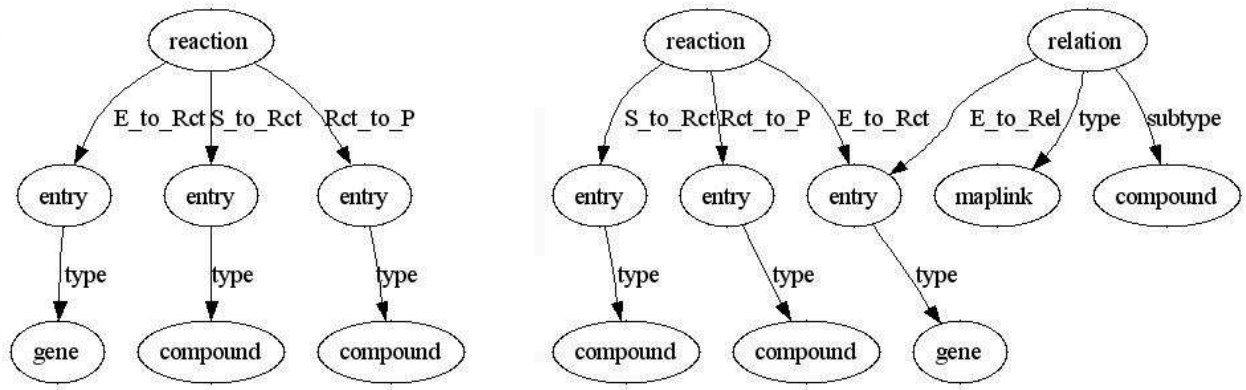



Figure 6.8. Running time with graph size of clustering in species.

If we have additional patterns to basic pattern (a), the pattern would have another biochemical meaning. Figure 6.9 (b) would be a good example. The best pattern in the 00010 biological network across all species has a basic enzyme-related reaction and a relation which has a maplink as its second entry. We run this pattern as a predefined substructure in the GDM phase 2 to know the biological meaning, and get the updated substructure in Figure 6.10. In the updated pattern the nodes and edges from Figure 6.9 (b) are checked with "[ ]", because the checked nodes and edges are from GDM phase 1. They are also shown in the Figure 6.9 (b). The pattern has two parts: one is the reaction, R02740 (shown in the Equation 6.4 and the Figure 6.11), and another part shows the relationship with other biological network, 00052, which is the Galactose Metabolism. Galactose, a type of sugar, is a part of lactose with glucose. Lactose, a sort of milk, is secreted from mammary glands [52].

Because the 00010 biological network, which is named Glycolysis, is an initial point of several pathways to generate biochemical energy, the products and metabolites of of glycolysis are used in a variety of other metabolism as starting or intermediate sub-




(a) First best pattern from 00061_45 set

(b) First best pattern from 00010_all set

Figure 6.9. The common best substructures in unsupervised learning.

strates. Metabolite is a intermediate in the metabolism. Moreover several enzymes are related with other metabolic pathways. Glycolysis in *Saccharomyces cerevisiae*(sce) has relationships with twenty-five other metabolic networks.

$$C_6H_{13}O_9P \leftrightarrow C_6H_{13}O_9 \tag{6.4}$$

$C_6H_{13}O_9P$ : alpha-D-Glucose 6-phosphate

$C_6H_{13}O_9P$ : beta-D-Fructose 6-phosphate

The pattern in figure 6.7 shows this relationship. CPD:C00668, alpha-D-Glucose 6-phosphate is used in Galactose metabolism, too. It is not simply sharing materials, but system. Alpha-D-Glucose 6-phosphate is a metabolite in both networks. If this compound is too sufficient in Glycolysis, Galactose metabolism tries to consume more. Or if there is too much in Galactose metabolism, Glycolysis is catalyzed to use more. Therefore the relation is needed at both biological networks. Unfortunately, the detailed

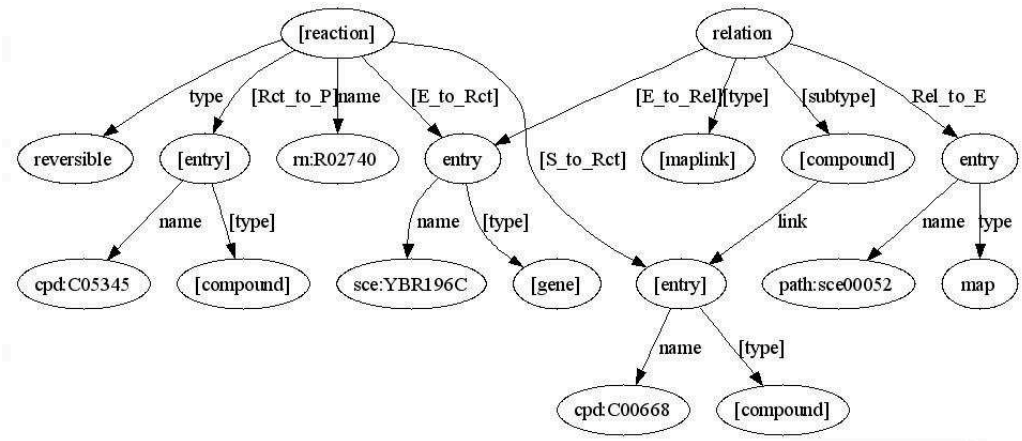



Figure 6.10. Updated first best substructure of 00010_all set.

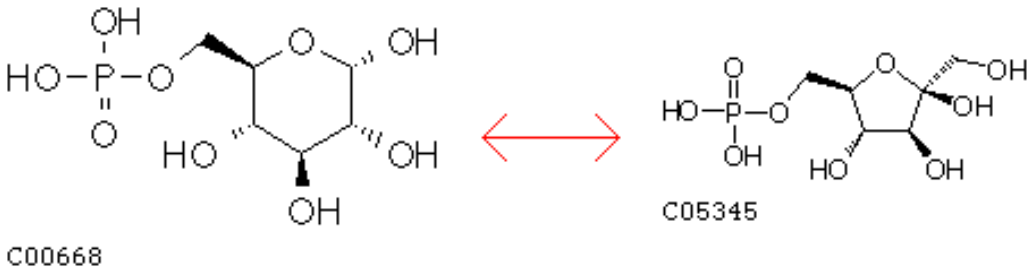



Figure 6.11. Reaction R02740 [2].

mechanism is not described in the KEGG pathway. Because we do not have the brilliant method to describe the regulation of the enzyme relation, it would be a challenge to many bioinformatics people.

### 6.2.2 Clustering in networks

Table 6.4 shows the result of clustering in networks. Set represents a name of species in a set. Other columns shows the same information as described above. In contrast with previous cases, the running time does not look polynomial. Especially, the average of running time in last four cases is 8 times longer than the one in first three cases. This vast

Table 6.4. Results of clustering in networks

| **Set** (Name of species) | **Positive examples** (Number) | **Size** (V+E) | **Iteration** (Number) | **Running time** (sec.) |
|---|---|---|---|---|
| ath | 102 | 69,668 | 10 | 810.56 |
| dme | 100 | 61,180 | 10 | 785.89 |
| eco | 103 | 78,543 | 10 | 1393.51 |
| rno | 120 | 76,608 | 10 | 34869.27 |
| sce | 90 | 65,353 | 10 | 44133.98 |
| mmu | 130 | 99,603 | 10 | 35023.34 |
| hsa | 139 | 112,176 | 10 | 52858.71 |

Table 6.5. Results of learned pattern in Figure 6.9 (a)

| **Set** (Name of species) | **Instances** (Number) | **Examples** (Number)($a$) | **Positive examples** (Number)($b$) | **Frequency** ($\frac{a}{b} \times 100$)(%) |
|---|---|---|---|---|
| ath | 1,261 | 97 | 102 | 95.09 |
| dme | 986 | 91 | 100 | 91.00 |
| eco | 1,466 | 99 | 103 | 96.11 |
| rno | 1,096 | 94 | 120 | 78.33 |
| sce | 1,091 | 84 | 90 | 93.33 |
| mmu | 1,545 | 104 | 130 | 80.00 |
| hsa | 1,725 | 107 | 139 | 76.98 |

gap can be explained with one aspect, the number of regulatory network. As described above, biological networks of KEGG PATHWAY can be categorized into two groups: Metabolic pathways and Regulatory pathways. Regulatory pathways include protein-protein interaction and regulatory networks. Currently, most of regulatory pathways are included into higher level species. Table 6.6 shows the number of regulatory pathways in each species. While metabolic network mainly contains enzymatic process, regulatory includes relation between two or more proteins and genes. Therefore structures of two networks are more or less different. We ran Subdue on the hsa set without any regulatory network. This set has 110 positive examples of metabolic networks. This run takes 960.13 seconds. In table 6.4 the entire set of hsa takes 52858.71 seconds. However,

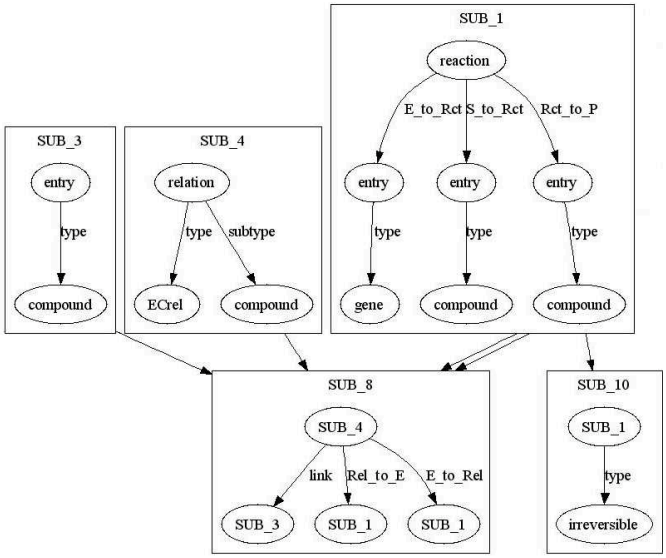



Figure 6.12. Parts of Hierarchical Clustering of biological networks in fruit fly.

*Saccharomyces cerevisiae*(sce) cannot be explained with this theory. We ran Subdue on the sce set in the same way as the hsa set. But the running time is similar. If we set Limit as 50, instead of 100, Subdue generates the same results in 375.96 seconds. This means that we can decrease running time with the same results by using Limit and Beam value as described in the chapter 4.

Table 6.6. Number of Regulatory networks

| **Set** (Name of species) | **Regulatory Pathways** (Number) |
|---|---|
| ath | 2 |
| dme | 8 |
| eco | 1 |
| rno | 24 |
| sce | 4 |
| mmu | 24 |
| hsa | 29 |

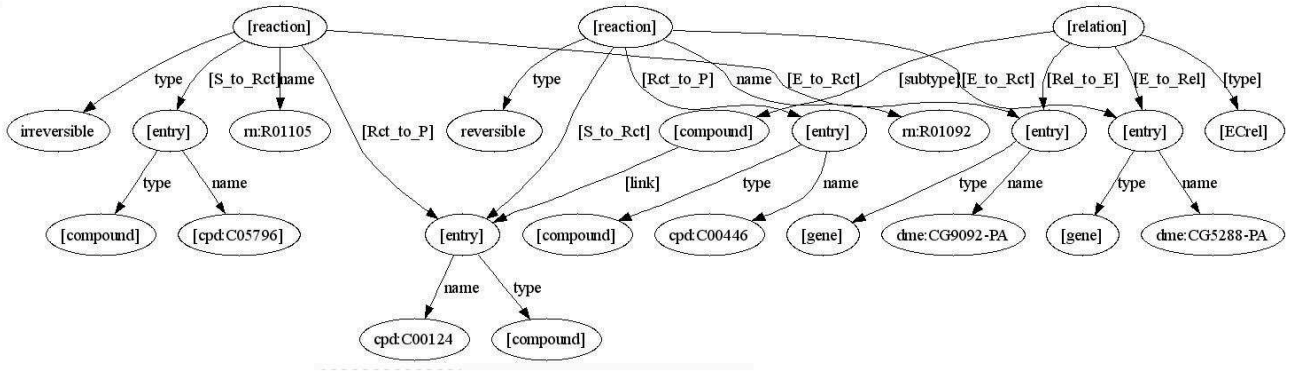



Figure 6.13. Updated eighth best substructure of Hierarchical Clustering of biological networks in fruit fly.

The approach of clustering in networks tries to find common patterns in all networks of a species. Several species are prepared and Subdue runs with limit 100 and MDL evaluation option for ten iterations. In the experimental sets Subdue finds the same best patterns as Figure 6.9 (a) at the first iteration. Table 6.5 shows the number of instance and examples in which the pattern in Figure 6.9 (a) is found. Frequency represents a probability of the pattern found in examples of a set.

The result from the dme set draws a hierarchical clustering tree of substructures in Figure 6.12. First best substructure, SUB_1 at the Figure 6.12, is the basic patterns of all networks in fruit fly. SUB_3 is found in 3,688 instances of 48 examples at the third iteration. SUB_4 found in 1,175 instances of 23 examples is the relation with ECrel property. ECrel relation is enzyme-enzyme relation that two enzymes catalyze successive reaction steps [3]. SUB_8 consists of two SUB_1, a SUB_3 and SUB_4 with several edges. This pattern found in 268 instances of 5 examples contains one relation of two enzymes which catalyze two successive reactions. Moreover, SUB_8 has one more meaning than SUB_4. The key is the link edge. As mentioned in the graph representation of chapter 6, a relation may have a link to compound as its subtype when it is ECrel type. The

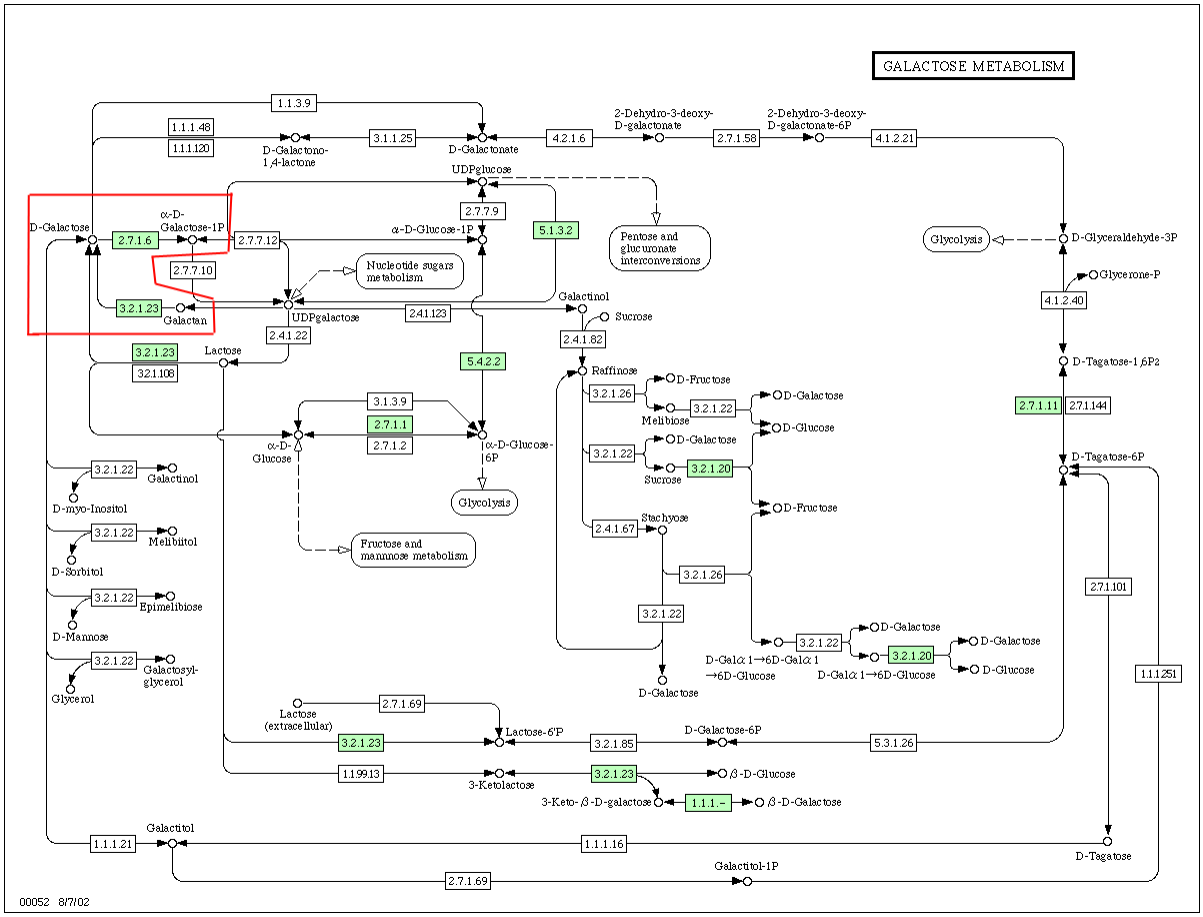



Figure 6.14. A graphic file map of Galactose metabolism in fruit fly [2].

link points to a compound which is a product of the first reaction of this relation and a substrate of second reaction at the same time.

Figure 6.13 shows an example of the pattern which is found in the 00052, Galactose metabolism, network of the fruit fly and updated by GDM phase 2. As in the same as previous example, the nodes and edges checked with "[ ]" are found at GDM phase 1, others are updated through GDM phase 2 manually. The enzyme-enzyme relation has a relationship of two reaction: R01092 (Figure 6.15) and R01105 (Figure 6.16). R01092 is catalyzed by the enzyme of the gene, dme:CG5288-PA, and R01105 is catalyzed by the enzyme of the gene, dme:CG9092-PA. The substrate of R01092 is the C05796 compound

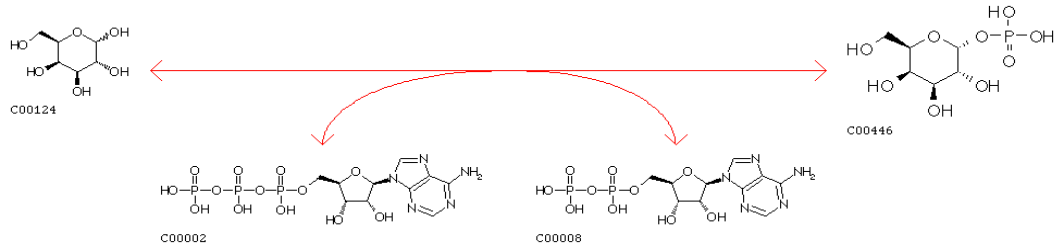



Figure 6.15. Reaction R01092 [2].

(Galactin). The product of this reaction is C00124 (D-Galactose) is also the substrate of R01092. R01092 produces C00446 (alpha-D-Galactose 1-phosphate) from the product compound. The relation of this pattern has the link as pointer to C00124, because this compound is the metabolite of two reactions of itself. The simple formula of those successive reactions is shown at equation 6.5. Figure 6.14 shows the pattern in a graphic file map of Galactose metabolism in fruit fly.

$$(C_{12}H_{20}O_{11})_n + ec : 3.2.1.23 \rightarrow C_6H_{12}O_6 + ATP + ec : 2.7.1.6 \leftrightarrow ADP + C_6H_{13}O_9P \quad (6.5)$$

$(C_{12}H_{20}O_{11})_n$ : galactin

$C_6H_{12}O_6$ : D-galactose

$C_6H_{13}O_9P$ : alpha-D-galactose 1-phospahte

$ec$ : 3.2.1.23 : beta-galactosidase from gene dme:CG5288-PA

$ec$ : 2.7.1.6 :galactokinase from gene dme:CG9092-PA

$ATP, ADP$ : biochemical energy materials

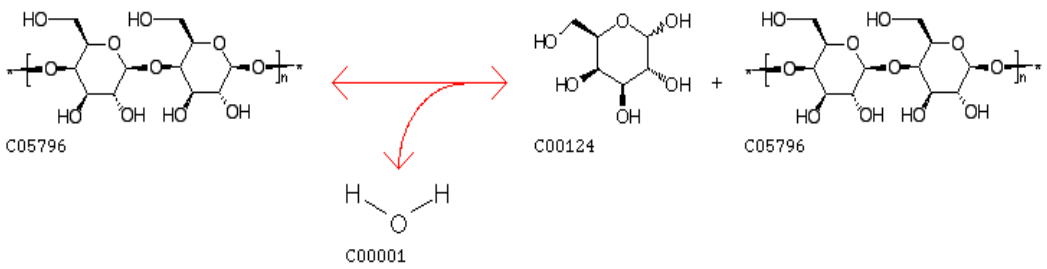



Figure 6.16. Reaction R01105 [2].

## 6.3 Summary

This chapter showed the results of our approaches to biological networks. In supervised learning Subdue can clearly distinguish two groups in the classification by networks. But it does not perform well in the classification by species because of the biochemical features of biological networks. Unsupervised learning of Subdue can find the best substructure in the both clustering in the same network across different species and in different networks of the same species. Moreover, the best substructures found by Subdue have important biological meaning. Additionally, we discussed the complexity issue in each section.

# CHAPTER 7

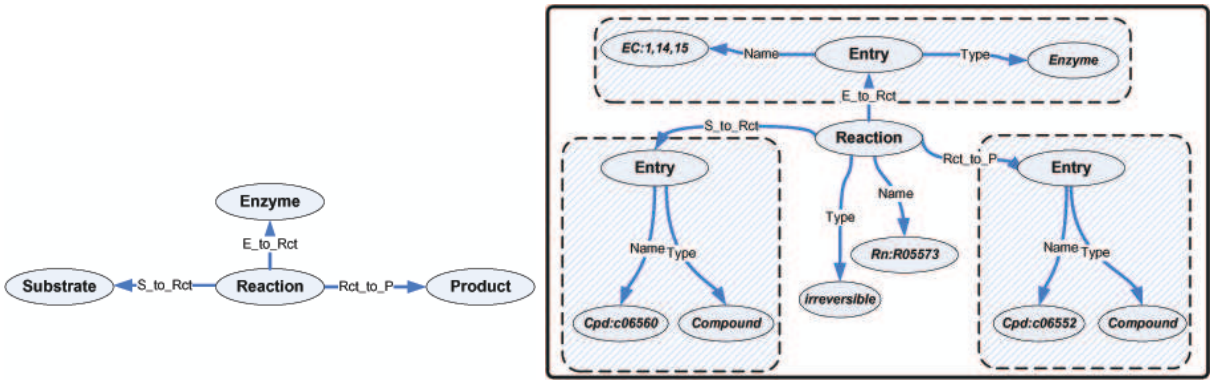
# CONCLUSION AND FUTURE WORK

## 7.1 Conclusion

The results of chapter 6 clearly show that the substructure found by Subdue has understandable biological meaning. The classification approach clearly discriminates two examples if they have sufficient features to distinguish. The clustering approach generates common substructures that allow us to better understand the biological network.

In this result we generate a graph representation of biological networks to describe all properties of the network. Then we apply our graph-based relational learning system, Subdue, to these graph representations. Unlike other graph-based data mining tools which are focusing on frequent patterns, the graph-based relational learning of the Subdue can focus on novel, useful and understandable graph-theoretic patterns. For this reason, our approach can discover the biologically meaningful pattern that is able to explain the meaning of the pattern and relationship with other substructures in the input graph based on the background knowledge.

Systems biology plays an important and meaningful role in bioinformatics, based on the huge amount of results which are accumulated by the extensive research in genomics, proteomics and other biochemical areas. To express a variety of objects in the network and complex relationships between objects, the graph representation is indispensable. Especially to understand bio-organisms systematically, knowledge discovery should be guided into finding biologically meaningful knowledge, not just frequent knowledge without special meaning.




(a) (b)

Figure 7.1. A graph representation of biological networks with nested graph concept, (a) abstract model and (b) extended model.

This research shows that the Subdue graph-based relational learning technique is applied successfully to the biological network domain. The substructures generated by Subdue can distinguish between two groups of biological networks or guide us to get more detailed knowledge of biological networks.

## 7.2 Future Work

Several challenges are available for future study. We categorize the challenges into two aspects: new discovery algorithm and research of biological networks.

### 7.2.1 Discovery algorithm

Subdue is successfully applied to biological networks in addition to other areas such as Chemical Toxicity [45], Molecular Biology [8], Security [6] and Web Search [7]. However, it is necessary to study continuously for better performance on more various areas. Basically, biological networks consist of relations and reactions. Relations and reactions have objects such as proteins and compounds in their attributes as well as their relationships. For this reason, biological networks can be minimized with major

objects (proteins) and their relationships [63]. But this approach misses several important features in biological networks. Even though our approach includes almost all features in biological networks, it also has some defects. First it does not show better performance because Subdue covers so many vertices and edges. Second it handles objects and their attributes in same manner. In our approach, entry vertex and its name vertex are managed as the same vertices. Semantically, the name vertex is an attribute of the entry vertex.

We introduce a new approach to Subdue not only for biological networks but also for bioinformatics and other areas which need to be represented as a hierarchical graph. Figure 7.1 shows an example of a graph representation of biological networks with the nested graph concept. It represents an abstract model (a) and an extended model. The abstract model provides a high level view of graphs. The extended model shows a lower level view inside each subgraph which is a vertex in the abstract model. Because our approach represents biological networks as a hierarchial graph, we can analyze biological networks in two levels: upper level (a) and lower level (b). For this representation, Subdue needs to be modified to find the patterns in two levels. By using this approach, Subdue can find three types of relational patterns. First, Subdue can learn the relational patterns on the entire graph in the same way as the current approach. Second, Subdue can find the abstract patterns in the entire graph using the abstract model. This method can provide a quick abstraction in the graphs. Third, local patterns can be found in each subgraph such as protein subgraphs, compound subgraphs and reaction subgraphs. The last method may not look useful on current data. But it can be a useful way to find the relational patterns with specific structures of objects in biological networks if the databases of biological networks can be integrated with a variety of data such as protein structures, gene sequences and compound structures.

This approach can be useful to other areas. In bioinformatics areas it can be applied to interactions between several cells. Cells have a variety of mechanisms intracellularly and intercellularly. We can represent their relation as hierarchical graphs. Also we can apply this approach to community ecology or social relationships between two communities. Because Subdue uses a general graph representation, more detailed research is necessary for application of our new approach.

### 7.2.2 Research of biological networks

There are several areas to better understand biological networks. First, more background knowledge need to be combined to describe the biological network. Even though the KEGG pathway database has sufficient information, it is not enough for system-level understanding of bio-organisms.

Second, an enhanced representation method is necessary to express all features of biological networks. Because biological data is redundant, knowledge representation is an unavoidable challenge. Specifically, temporal and spatial concepts should be included into the representation. The regulatory network plays a central role to maintain our body in a peaceful and optimal state. The study of the regulatory network without concerning dynamics is scarcely ever helpful to understand.

Third, background knowledge needs to be included into graph-based relational learning. Even though Subdue can use the background knowledge as the predefined substructure, it is still to be enhanced. If more knowledge is included into the learning algorithm, more meaningful results can be generated. In this way, the database of the substructures of biochemical functional groups will be helpful in addition to protein, gene and compound databases.

## REFERENCES


[1] L. Luscombe, D. Greenbaum, and M. Gerstein, "What is bioinformatics? an introduction and overview," *Yearbook of Medical Informatics*, pp. 83–100, 2001.

[2] Kanehisa Laboratories. KEGG pathway website. [Online]. Available: http://www.genome.jp/kegg/pathway.html

[3] ——. KEGG Markup Language manual. [Online]. Available: http://www.genome.jp/kegg/docs/xml/

[4] J. Watson and F. Crick, "Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid," *Nature*, no. 4356, pp. 737–738, Apr. 1953.

[5] S. Stephens, J. Rung, and X. Lopez, "Graph data representation in oracle database 10: Case studies in life sciences." *IEEE Data Eng. Bull.*, vol. 27, no. 4, pp. 61–66, 2004.

[6] L. Holder, D. Cook, J. Coble, and M. Mukherjee, "Graph-based relational learning with application to security," *Fundamenta Informaticae Special Issue on Mining Graphs*, vol. 6.

[7] D. J. Cook, N. Manocha, and L. B. Holder, "Using a graph-based data mining system to perform web search," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 5, 2003.

[8] S. Su, D. Cook, and L. Holder, "Application of knowledge discovery to molecular biology: Identifying structural regularities in proteins," in *Pacific Symposium on Biocomputing*, vol. 4, 1999, pp. 190–201.

[9] M. Chen, J. Han, and P. Yu, "Data mining: an overview from a database perspective," *IEEE Trans. Electron. Comput.*, vol. 8, pp. 866–883, 1996.

[10] M. K. J. Han, *Data Mining, Concept and Techniques*. NY: Morgan Kaufmann, 2001.

[11] B. Bergeron. (2002, Nov.) Applied bioinformatics computing: An introduction. [Online]. Available: http://www.informit.com/articles/article.asp?p=30121

[12] D. Page and M. Craven, "Biological applications of multi-relational data mining," *ACM SIGKDD Explorations Newsletter*, vol. 5, pp. 69–79, July 2003.

[13] B. Bergeron. (2002, Dec.) Applied bioinformatics computing: Data mining. [Online]. Available: http://www.informit.com/articles/article.asp?p=30169

[14] D. Benson and et al, "GenBank," *Nucleic Acids Research*, vol. 33, pp. 34–38, 2005.

[15] C. Kanz and et al, "The EMBL Nucleotide Sequence Database," *Nucleic Acids Research*, vol. 33, pp. 29–33, 2005.

[16] S. Miyazaki and et al, "DDBJ in the stream of various biological data," *Nucleic Acids Research*, vol. 32, pp. 31–34, 2004.

[17] A. Bairoch and et al, "The Universal Protein Knowledgebase," *Nucleic Acids Research*, vol. 33, pp. 154–159, 2005.

[18] N. Hulo and et al, "Recent improvements to the PROSITE database," *Nucleic Acids Research*, vol. 32, pp. 134–137, 2004.

[19] M. eshpande, M. Kuramochi, N. Wale, and G. Karypis, "Frequent substructure-based approaches for classifying chemical compounds," *IEEE Trans. Knowledge Data Eng.*, vol. 17, no. 8, pp. 1036–1050, 2005.

[20] R. Geer and et al, "Entrez: Making use of its power," *Briefings in* B*ioinformatics*, vol. 4, pp. 1779–1784, 2003.

[21] R. Dowell, R. Jokerst, A. Day, S. Eddy, and L. Stein, "The Distributed Annotation System," *Bioinformatics*, vol. 2, 2001.

[22] R. Edgar, M. Dormrachev, and A. Lash, "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository," *Nucleic Acids Research*, vol. 30, pp. 207–210, 2002.

[23] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J. Molecular Biology*, vol. 147, pp. 195–197, 1981.

[24] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Molecular Biology*, vol. 48, pp. 443–453, 1970.

[25] O. Gotoh, "An improved algorithm for matching biological sequences," *J. Molecular Biology*, vol. 162, pp. 705–708, 1982.

[26] S. Altschul, W. Gish, E. M. W. Miller, and D. Lipman, "Basic local alignment search tool," *J. Molecular Biology*, pp. 403–410, 1990.

[27] S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, pp. 3389–3402, 1997.

[28] J. Thompson, D. Higgins, and T. Gibson, "Clustal w: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting,position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, pp. 4673–4680, 1994.

[29] D. Lipman, S. Altschul, and J. Kececioglu, "A tool for multiple sequence alignment," in *Proc. the National Academy of Sciences*, vol. 86(12), June 1989.

[30] A. Krogh, *Computational Methods in Molecular Biology.* Elsevier, 1998, ch. An Introduction to Hidden Markov Models for Biological Sequences, pp. 45–63.

[31] M. Kurella, L. Hsiao, T. Yoshida, J. Randall, G. Chow, S. Sarang, R. Jensen, and S. Gullans, "Dna microarray analysis of complex biologic processes," *J. American Society of Nephrology*, vol. 12, pp. 1072–1078, 2001.

[32] H. Mamitsuka, Y. Okuno, and A. Yamaguchi, "Mining biological active patterns in metabolic pathways using microarray expression profiles," *ACM SIGKDD Explorations Newsletter*, vol. 5(2), pp. 113–121, Dec. 2003.

[33] G. Piatetsky-Shapio and P. Tamayo, "Microarray data mining: Facing the challenges," *ACM SIGKDD Explorations Newsletter*, vol. 5(2), pp. 1–5, Dec. 2003.

[34] H. Kitano, *Foundations of Systems Biology*. MIT Press, 2001, ch. Systems Biology: Toward System-level Understanding of Biological Systems, pp. 1–36.

[35] S. Dzerosk, "Multi-relational data mining: an introduction," *ACM SIGKDD Explorations Newsletter*, vol. 5, pp. 1–16, July 2003.

[36] M. Sternberg and S. Muggleton, "Structure activity relationships (SAR) and pharmacophore discovery using inductive logic programming (ILP)," *QSAR and Combinatorial Science*, vol. 22, 2003.

[37] S. Muggleton, "Machine learning for systems biology," in *Proceedings of the 15th International Conference on Inductive Logic Programming*, 2005, pp. 416–423.

[38] H. Lodhi and M. S, "Modelling metabolic pathways using logic programs-based ensemble methods," in *Computational Methods in Systems Biology*, 2005.

[39] N. Jones and P.Pevzner, *An Introduction to Bioinformatics Algorithms*. The MIT Press, 2004, ch. Graph Algorithm, pp. 247–310.

[40] L. B. Holder and D. J. Cook, *A Encyclopedia of Data Warehousing and Mining*. Idea Group Publishing, 2005, ch. Graph-based Data Mining, pp. 247–310.

[41] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *IEEE Conference on Data Mining*, 2001, pp. 313–320.

[42] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *IEEE Conference on Data Mining*, 2002.

[43] L. B. Holder and D. J. Cook, "Graph-based relational learning: current and future directions," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 1, pp. 90–93, 2003.

[44] D. J. Cook and L. B. Holder, "Graph-based data mining," *IEEE Intelligent Systems*, vol. 15, no. 2, pp. 32–41, 2000.

[45] R. Chittimoori, L. B. Holder, and D. J. Cook, "Applying the subdue substructure discovery system to the chemical toxicity domain," in *Proceedings of the Florida AI Research Symposium*, 1999, pp. 90–94.

[46] D. J. Cook and L. B. Holder, "Substructure discovery using minimum description length and background knowledge," *Journal of Artificial Intelligence Research*, vol. 1, pp. 231–255, 1994.

[47] P. Grünwald, *Advances in Minimum Description Length.* the MIT Press, 2005, ch. A tutorial introduction to the minimum description length principle, Chapter 1, 2.

[48] H. Bunke and G. Allerman, "Inexact graoh matching for strucutral pattern recognition," *Pattern Recognition Letters*, vol. 1, no. 4, pp. 245–253.

[49] D. J. Cook, L. B. Holder, and S. Djoko, "Scalable discovery of informative structural concepts using domain knowledge," *IEEE Expert: Intelligent Systems and Their Applications*, vol. 11, no. 5, pp. 59–68, 1996.

[50] J. A. Gonzalez, L. B. Holder, and D. J. Cook, "Graph-based relational concept learning," in *Proceedings of International Conference on Machine Learning*, 2002, pp. 219–226.

[51] L. B. Holder, D. J. Cook, J. Gonzalez, and I. Jonyer, *Pattern Recognition and String Matching.* Springer, 2003, ch. Structural Pattern Recognition in Graphs, pp. 255–280.

[52] D. Nelson and M. Cox, *Lehninger Principles of Biochemistry.* W.H. Freeman and Company, 2005.

[53] L. Hunter and et al., *Artificial Intelligence and Molecular Biology.* AAAI Press, 1993, ch. Molecular Biology for Computer Scientists, pp. 1–46.

[54] J. Berg, J. Tymoczko, and L. Stryer, *Biochemistry.* W. H. Freeman and Co., 2002.

[55] P. D. Karp and M. L. Mavrovouniotis, "Representing, analyzing, and synthesizing biochemical pathways," *IEEE Expert: Intelligent Systems and Their Applications*, vol. 9, no. 2, pp. 11–21, 1994.

[56] H. K. O. Wolenhauer and and K. Cho, "Systems biology," *IEEE Control Syst. Mag.*, vol. 15, pp. 38–48, Aug. 2003.

[57] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, pp. 1662–1664, 2002.

[58] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems Biology*, 1st ed. WILEY-VCH, 2005.

[59] C. Schilling, S. Schuster, B. Pallson, and R. Heinrich, "Metabolic pathway analysisl basic concepts and scientific applications in the post-genomic era," *Biothcnology Progress*, vol. 15, pp. 296–303, 1999.

[60] K. Cho and O. Wolenhauer, "Analysis and modelling of signal transduction pathways in systems biology," in *Biochemical Society Transactions*, vol. 31, 2003, pp. 1503–1509.

[61] C. Yoo, V. Thorsson, and G. Cooper, "Discovery of causal relationships in a gene regulation pathway from a mixture of experimental and observational DNA microarray data," in *Pacific Symposium on Biocomputing*, vol. 7, 2002, pp. 498–509.

[62] M. Kanehisa, S. Goto, S. Kawashima, U. Okuno, and M. Hattori, "Kegg resource for deciphering the genome," *Nucleic Acids Research*, vol. 32, pp. 277–280, 2004.

[63] M. Koyuturk, A. Grama, and W. Szpankowski, "An efficient algorithm for detecting frequent subgraphs in biological networks," *BIOINFORMATICS*, vol. 20, pp. 200–207, 2004.

[64] H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou, "Mining coherent dense subgraphs across massive biological networks for functional discovery," *Bioinformatics*, vol. 21, no. 1, pp. 213–221, 2005.

[65] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabasi, "The large-scale organization of metabolic networks," *Nature*, vol. 407, 2000.

[66] A. Vázquez, R. Dobrin, J. E. D. Sergi and, Z. N. Oltvai, and A. Barabási, "The topological relationship between the large-scale attributes and local interaction patterns of complex networks," *Proc. the National Academy of Sciences*, vol. 101, no. 52, pp. 17 940–17 945, 2004.

[67] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker, "Conserved patterns of protein interaction in multiple species," *Proc. the National Academy of Sciences*, vol. 102, pp. 1974–1979, 2004.

[68] H. Kitano, "A graphical notation for biochemical networks," *BIOSILICO*, vol. 1, pp. 169–176, 2003.

[69] H. Kitano, A. Funahashi, Y. Matsuoka, and K. Oda, "Using process diagrams for the graphical representation of biological networks," *Nature Biotechnology*, vol. 23, pp. 961–966, 2005.

[70] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of escherichia coli," *Nature Genetices*, vol. 31, pp. 64–68, 2002.

[71] P. Kharchenko, G. M. Church1, and D. Vitkup, "Expression dynamics of a cellular metabolic network," Tech. Rep.

[72] G. Bader, D. Betel, and C. Hogue, "Bind:the biomolecular interaction network database," *Nucleic Acids Research*, vol. 31, pp. 248–250, 2003.

[73] C. Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel, "String:a database of predicted functional associations between proteins," *Nucleic Acids Research*, vol. 31, pp. 258–261, 2003.

[74] G. Bader, D. Betel, and C. Hogue, "Pathways database system: an integrated system for biological pathways," *Bioinformatics*, vol. 19, no. 2, pp. 930–937, 2003.

[75] (2003) Extensible markup language (xml). [Online]. Available: http://www.w3.org/XML/

[76] (2004, Apr.) Document object model (dom) level 3 core specification. [Online]. Available: http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/

[77] McLaughlin, *Foundations of Systems Biology.* O'Reilly, 2001.

## BIOGRAPHICAL STATEMENT

Chang hun You was born in Seoul, Korea in 1976. He received B.A. in Agricultural Biology from The Korea University in 2002. He began his study toward the M.S. degree in the department of Computer Science and Engineering at The University of Texas at Arlington in August 2003. His research interests focus on data mining and bioinformatics.