TEXT MINING BIOMEDICAL LITERATURE FOR CONSTRUCTING GENE REGULATORY NETWORKS

By

YONGLING SONG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2007

To my family

ACKNOWLEDGMENTS

I acknowledge the assistance of my advisor Dr. Su-Shing Chen. I appreciate his valuable academic advice, his kindness, help and support without which, it would have been impossible for me to assume the process toward my Ph.D. degree. I would like to thank all of my Ph.D. committee members---Dr. Randy Chow, Dr. Jih-Kwon Peir, Dr. Shouguang Jin, and Dr. Chris Jermaine for their kindness and valuable advice on my research. I would like to thank Dr. Randy Chow for his kindness and help on my TA work. I enjoy working with Dr. Chow and appreciate all the help that he has provided. I also would like to thank Dr. Jih-Kown Peir. As the graduate student coordinator, Dr. Peir provided a lot of help and advice on my Ph.D. work. I thank Dr. Chris Jermaine, whose teaching I have enjoyed immensely and I have learnt a lot from his course. I would like to thank Dr. Shouguang Jin for providing me with this interesting research topic and continually helping me on it.

I want to thank all of my colleagues, especially Weihui Wu for his help and collaboration and Hyunki Kim for all the useful advice on my research.

Most of all I would like to thank my husband, Yuan Guo, for his continued encouragement and help during these years.

TABLE OF CONTENTS

page

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

TEXT MINING BIOMEDICAL LITERATURE FOR CONSTRUCTING GENE REGULATORY NETWORKS

By

Yongling Song

August 2007

Chair: Su-Shing Chen
Major: Computer Engineering

As the result of decades of active research, tremendous amount of experimental data are available on gene regulatory networks. The ability to dynamically visualize the complex data and logically integrate the knowledge related to gene function, gene regulation, and biomedical evidence would be useful for individual researchers to keep up with all the information and would provide a global view about gene regulatory networks.

In our research, we present the framework of a gene regulatory networks system (GRNS). GRNS automatically mines biomedical literature to extract gene regulatory information (strain number, genotype, gene regulatory relation, and phenotype), automatically constructs gene regulatory networks based on extracted information, and integrates biomedical knowledge into the regulatory networks.

First, GRNS uses an automated text mining technique to extract information about regulatory networks from the collection of biomedical texts. GRNS extracts five kinds of gene regulatory information: strain number, genotype, gene regulatory relation, phenotype, and unrecognized sentence. Based on the extracted gene regulatory information, GRNS can automatically construct and visualize gene regulatory networks. Second, to provide researchers with a clear and global view about the regulatory networks, GRNS uses an interactive

visualization method to integrate biomedical evidential information into the regulatory networks. Once a user clicks an entity or a relation of interest, the query interface returns a detailed information page about the clicked entity or relation. GRNS logically integrates the knowledge related to gene function, gene regulation and biomedical evidences, collects genetic evidences, biochemical tests, sequence based predications or biomedical literatures and links this information with regulatory relationships and regulatory entities data. Third, GRNS provides analysis tools for gene regulatory networks. The analysis tools include the frequent graph mining tool and the gene relation predication tool.

CHAPTER 1
INTRODUCTION

Motivation

After decades of active research, tremendous amounts of experimental data are available on the gene function and their regulation in different genomes (Shatkay, *et al.*, 2003; Cohen, *et al.*, 2004; Goodman, *et al.*, 2004; Woods, *et al.*, 2004). The experimental data is embedded in tens of thousands of published literature and it is difficult for the individual researcher to extract a comprehensive view of the gene function and regulation in different genomes. Research on different genomic sequences within recent years has resulted exponential growth of related literature (Cohen, *et al.*, 2005; Hirschman, *et al.*, 2002). To help individual scientists to keep up with all the new information, a complete system that not only compiles the experimental evidences but also logically integrates the knowledge related to gene function and regulation is desired.

Background

"A gene regulatory network is a collection of DNA segments in a cell which interact with each other and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA" (Jong, *et al.*, 2002). The regulation can be at different levels --- the transcriptional level (activation or repression), the post transcriptional level (mRNA stability), and the translational or post-translational level (protein-protein interaction/modification) (Fig 1-1). These gene regulatory networks provide researchers with a global view on the function of genes and the relationships among several regulatory elements and facilitate the acquisition of relative information and design of future experiments. There are several ways to construct and simulate gene regulatory networks. We can construct gene

regulatory networks by the experimental method, by the mathematical models method, by the rule-based knowledge method, or by the text mining biomedical literature method.

Experimental Method

Using the experimental method is the most direct way to find the regulatory relations between genes/proteins. A lot of biologists exploit this path to find out the regulation information. The experimental method is also the most accurate way among all the methods to construct gene regulatory networks. However, the experimental method is usually employed in finding the relation information among a small number of genes. It is very difficult to find the regulation relations in a large-scale genome data by this method. The experimental method usually provides us accurate but a small-scale of regulatory relation information.

Mathematical Models for Regulatory Networks Construction

The recently developed cDNA microarray technology not only provides a large amount of available gene expression data, but also provides the possibility of using mathematical models to simulate gene expression data and construct regulatory networks. Various mathematical modeling techniques have been used, including Boolean networks, Probabilistic Boolean Networks, Bayesian networks, and other models.

Boolean networks model

The Boolean network model was introduced by Kauffman (Kauffman, 1969) and recently developed by Shmulevich (Shmulevich, *et al*., 2002). Interactions between genes can be represented by Boolean functions that calculate the state of a gene from the activation of other genes. In the Boolean network model, gene expression only has two levels: ON and OFF. For a gene, "ON" corresponds to the gene being expressed; for inputs and outputs, "ON" corresponds to the substance being present. Boolean gene expression networks represent a relatively simple

model of the gene expression control mechanisms of the cell. However, this model has not achieved any interesting theoretical results yet.

Probabilistic boolean networks model

Another classical model is Probabilistic Boolean Networks (PBNs) (Shmulevich *et al*., 2002), which extends the Boolean network to accommodate more than one possible function for each node and has the ability to handle data and the model selection uncertainty. PBNs combine several promising Boolean functions to predict a target gene. The Markov Chain theory can be used to analyze the dynamics of PBNs.

Bayesian networks model

"A Bayesian network (also known as causal probabilistic networks) is an annotated directed acyclic graph that encodes a joint probability distribution of a set of random variables" (Friedman *et al*., 2000). The features of the Bayesian network model include, providing clearly depicting dependencies and independencies between variables, providing a probability distribution and combining machine learning and uncertainty. However, the strictly mathematical assumptions of the Bayesian network model do not always hold in gene expression data.

All of these mathematical models, Boolean networks, Probabilistic Boolean Networks or Bayesian networks, have some promising features in regulatory networks construction. But almost all of these mathematical models need strictly mathematically assumptions and these assumptions may not hold in gene expression data. Besides, the constructed regulatory networks are difficult to validate. Many mathematical models are developed to simulate the gene expression data and construct regulatory networks. It is still uncertain if they are good enough to present the gene expression data.

Rule-based Method for Regulatory Networks Construction

Another method to regulatory networks construction is the knowledge-based or the rule-based simulation method (Brutlag. *et al*., 2000). Basically, a rule-based method has two components, a set of facts and a set of rules stored in a knowledge base. The rule-based method can deal with a richer variety of biological knowledge. However, it has the difficulties in maintaining the consistency of a knowledge base and incorporating quantitative information. Although there are some attempts to use this method, it is not as popular as other methods.

Text Mining Biomedical Literature for Regulatory Networks Construction

Another method to construct regulatory networks is by text mining biomedical literature. A text mining system searches for entities, relationships among them, or other specific objects within text documents (Nenadic, *et al*., 2003; Liu, *et al*., 2003; Afantenos, *et al*., 2005; Yandell, *et al*., 2002). Each document is processed to identify and/or extract possible meaningful and content bearing entities and relationships. This text mining method has several advantages over other methods:

1. It can process a large set of biomedical literature. Therefore, it can construct regulatory networks representing large-scale genome data.

2. It extracts evidential information from biomedical literature. Biomedical literature usually records experimental methods and evidence in discovering the gene regulation relations. The experimental method is still the most accurate way to find gene regulation relations. Therefore, if we can extract the information from biomedical literature correctly, we can get the most accurate regulation relations information as well. The correctness of the regulation relations information in the text mining method is better than other mathematical models or rule-based method.

3. There is a large amount of biomedical literature available. For example, the PubMed database contains information for over 12 million articles and continues to grow at a rate of 2000 articles per week. The text mining method has enough resources to extract the useful information and construct gene regulatory networks.

We have decided that the text mining is the best method to construct the gene regulatory networks, for text mining is based on the existing experiment results. All other modeling

methods use mathematical or computational models which may be unrealistic. These models have not been proven that they can totally represent the biological simulation the processes. They have used a lot of assumptions, which are usually not always true in real biological world. But the text mining method uses the result in the biomedical literature , which has been accumulated throughout the years by biologists. The biomedical literature describes the gene regulatory information based on the real experiment results. The experiment method has been proven that it is most direct way to find the gene regulatory relation. In the other hand, the text mining method summarizes gene regulatory network for a large set of genes and species, which the experiment results have accumulated. The text mining method is useful only when it achieves high precision in data extraction. Only highly precise data can provide the ability to construct useful gene regulatory networks. In this dissertation, we provide a framework of the text mining method to construct gene regulatory networks.

Objective

Our objective was to extract the gene regulatory information from a large biomedical text collection, then constructed and analyzed the gene regulatory networks based on the information we retrieve from the biomedical text collection. To achieve this goal, we built the gene regulatory networks system (GRNS for short). First, the GRNS can automatically extract the gene regulatory information from a collection of unstructured biomedical text; second, based on extracted information from the biomedical text, the GRNS can construct and visualize regulatory networks; third, the GRNS can analyze regulatory networks data, for example, finding a frequent pattern and predicting new gene regulation relations. To achieve the goal of GRNS, the text mining and visualization techniques are used.

Approach and Related Work

In recent years, the extraction of knowledge from biological literature has received considerable attention. For example, Blaschke et. al. (Blaschke *et. al*., 1999) used the statistical "bag of words" approach to the extraction protein-protein interaction. Bengio, Y. (Bengio, Y. 1993) build a Memory-based learning (MBL) system to do the text mining. Yakushiji et. al. (Yakushiji *et. al.*, 2001) designed an information extraction system using a general-purpose full parser. Friedman et. al. (Friedman *et. al.*, 2001) presented a GENIES system which extracts structured information about cellular pathways from biomedical literatures. Marcotte et. al. (Marcotte *et. al.*, 2001) showed a Bayesian approach of mining literature. McDonald et. al. (McDonald *et. al.,* 2004) developed an Arizona Relation Parser for extracting gene pathway relations. Chun et. al. (Chun *et. al.*, 2005) introduced a system to extract disease-gene relations from Medline by using a dictionary matching with machine learning-based named entity recognition approach. Hu. et. al. (Hu. *et. al.*, 2005) developed a rule-based system RLIMS-P to do the database annotation of protein phosphorylation. Yuan et. al. (Yuan *et. al.*, 2006) developed a web-based version of RLIMS-P. Saric et. al. (Saric *et. al.*, 2005) presented a rule based approach for extracting information from biomedical text.

There are two most used methods to extract biological knowledge: either a statistical method based on co-occurrences of proteins or genes, or a rule-based extraction method. Statistical methods are good at locating potential protein-protein interactions. But, they usually cannot provide a clear classification of interaction information. Rule-based relation extraction methods can achieve good precision and recall if the manually developed pattern is good. For example, Hu et. al. developed very good pattern templates to extract protein phosphorylation information. But it is difficult to build a set of complete pattern templates even for a biological expert. In this dissertation, we provide a framework of a rule-based method with the help of

potential informative sentences discovering. Therefore, new templates and rules can be incrementally supplemented.

To help researchers understand the complex regulatory networks, a good visualization tool is necessary. The visualization of gene regulatory networks is another open field in analysis of biological networks. Only recently, various bioinformatics tools became available for visualization regulatory networks or metabolic pathways, such as Pathways Database System-http://nashua.cwru.edu/pathways (Krishnamurthy *et al.*, 2003) ; VitaPad: visualization tools for the analysis of pathway data -http://bioinformatics.med.yale.edu (Holford *et al.*, 2005); BioMiner—modeling, analyzing, and visualizing biochemical pathways and networks--http://www.zbi.uni-saarland.de/chair/projects/BioMiner (Sirava *et al.*, 2002); The Pathway Tools Software (Karp *et al.*, 2002); Graphical interface to the gene network database GeNet-http://www. csa.ru/Inst/gorb_dep/inbios/genet (Serov *et al.*, 1998); PATIKA web: a Web interface for analyzing biological pathways through advanced querying and visualization-http://web.patika.org (Dogrusoz *et al.*, 2006). These bioinformatics tools are good at pathway visualization or pathway information query. However, these visualizations or query tools are not sufficient for researchers to understand the complex relationships among various biological subsystems. In the GRNS, we provide an interactive way to browse regulatory networks. Users can not only browse the networks but also access the related regulatory entities, the biomedical evidences by simply clicking a regulatory entity or regulatory relationship on the networks. Once a user clicks on the entity or regulatory relationship, a query interface returns a detailed information page to the user. Moreover, for every query, GRNS automatically generates a visualized result, giving the user a more clear view about relationships of the query entities. In

this way, GRNS logically integrates the knowledge related to gene function, gene regulation and biomedical evidence.

To help researchers analyze the regulatory networks, some analysis tools are needed. For example, a tool for mining the biologically meaningful patterns is useful to discover common motifs of cellular interaction and evolutionary relationships (Lawrence *et al.*, 1993). We model regulatory networks by directed graphs, which represent the entry elements in vertices and the relations between entry elements as directed edges. Now we can convert mining problems in regulatory network into graph mining problem. In general, solving the frequent graph mining problem in a large graph database is not an easy task. Subgraph isomorphism, already known as an NP-complete problem, plays an important role in the computation of frequent subgraphs mining. To reduce this problem, we use the simplifications of utilizing unique vertex labels. With this simplification, we reduce the frequent graph mining problem to the frequent item sets mining problem. Item sets problem is a well-studied problem in data mining field, and there are several well-known algorithms to deal with the frequent item set mining problem. Among them, Apriori (Agrawal *et al.*, 1994, 1995) is one of the most efficient algorithms. In recent years, based on the idea of reducing normal graph mining to frequent item set mining, several such algorithms have appeared (Koyutürk *et al.*, 2004; Kuramochi *et al.*, 2004; Inokuchi *et al*., 2003; Sartipi *et al*.,2001). But none of these algorithms show how to deal with inexact match graph mining. Inexact match graph mining is important because in most real applications (especially in the regulatory networks), due to the effects of noise, distortion, sampling error, or lack of known fixed patterns, many of the interesting subgraphs would show up slightly different. In this dissertation, we provide some heuristic methods to deal with inexact matching graph mining.

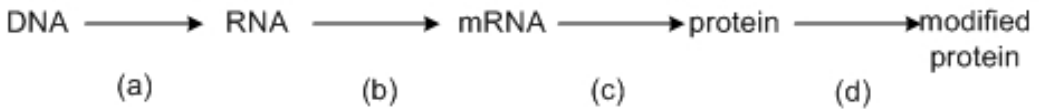
Research Contributions

The contributions of my research are three-fold.

First, we propose a rule-based text mining method to extract gene regulatory information by analyzing full-length biomedical text. The gene regulatory information includes gene regulatory relation information, strain number, genotype and phenotype information. Most text mining biomedical literature systems only provide methods to extract gene relation information. We use the strain table content extraction method to extract the strain number and genotype. We provide supplement rules to extract phenotype information. Strain number, genotype, and phenotype information provide supplemental evidence information for extracted gene regulatory relation information and help the researcher understand and validate the extracted gene relation information. We also provide a method to discover potential informative sentences. Experts can find potential new rules based on the unrecognized but informative sentences. This method is useful since it is difficult for a system to provide complete rules and templates sets in the beginning.

Second, we provide automatic constructs gene regulatory networks methods based on graphical representation schemas and graph visualization algorithms. Vertices and edges in regulatory networks link to the related phenotypes of gene mutants information. Phenotypes of gene mutants are recorded directly from the results of published papers. The function and regulatory role of each gene is based on the conclusions of the literatures and supplemented with sequence-based predictions. Moreover, for every query, we automatically generate visualized results, giving the user a clear view about the query entities. In this way, GRNS logically integrates the knowledge related to gene function, gene regulation, and biomedical evidence. Most other systems provide visualization of gene regulatory network, but do not link it to relate information in an interactive way and lack the ability of automatically generating the visualization result for every query result.

Third, we not only extract gene relation information from the biomedical text, we also provide analysis tools for the extracted information. The frequent pattern mining tool in GRNS extend some previous work on graph mining to inexact match graph mining which is important for real application. It also provides some heuristic methods to reduce the computation time comparing with other inexact algorithms. The gene relation prediction tool can predict potential gene regulatory relations based on current gene relation information in the database and prediction algorithm.



(a) transcriptional level

(b) post transcriptional level

(c) translational level

(d) post translational level

Figure 1-1. Regulation of gene expression at different levels. a) transcriptional level b) post transcriptional level c) translational level d) post translational level.

CHAPTER 2
SYSTEM OVERVIEW

In this chapter, we present the GRNS system overview, including the system architecture of GRNS, the data modeling of GRNS, the visualization modeling of regulatory networks in GRNS and the description of the experimental dataset-- *Pseudomonas aeruginosa* Genome.

System Architecture of GRNS

The system architecture of GRNS is shown in Figure 2-1. The GRNS employs two kinds of Database - Local Database and External Database. The Local DB of GRNS consists of five kinds of databases. Gene Database (GDB) stores the basic information about genes, proteins and products. English Dictionary DB stores information about English words, including synonyms. Rule DB stores information about text mining rules and templates information. Regulatory Network Database (NDB) stores the information about regulatory networks, such as subsystem information and regulatory relation information. Reference Information Database (RDB) stores the information about reference papers and biomedical evidence, such as genotypes, phenotypes, and strain names. The external DB resource includes some general database resources, such as GeneBank[1] and SWISS-PROT (Boeckmann, *et al*., 2003). It also includes some specific organism resources, for example, *Pseudomonas aeruginosa* Genome Database[2] (Stover, *et al*., 2000) for *Pseudomonas aeruginosa* Genome.

GRNS consists of three subsystems -- Text Mining Biomedical Literature Subsystem, Online Tools Subsystem, and Analysis Tools Subsystem. The structure of Text Mining Biomedical Literature Subsystem consists of four modules, Knowledge Collection and the Creation Module, the Pre-Processing Module, the Information Extraction Module (IE for short),

[1] http://www.ncbi.nih.gov/Genbank/

[2] http://www.pseudomonas.com/

and the Post-Processing Module. The tasks of the Knowledge Collection and the Creation Module include: first, integrating external database data into the local database; second, creating the templates and rules knowledge for the Information Extraction Module. The Pre-Processing Module downloads the full-length biomedical texts from PubMed Database and sends the corpus of texts to the IE Module. First the IE Module does strain-number and genotype information extraction. Then IE Module does the tokenization, sentences splitter and Part of Speech tagging. Then the IE Module recognizes the gene, the protein entities, and discriminating words from the corpus of texts. Then it extracts the relation, the phenotype, and other kinds of entities based on a rule-based approach. Finally, the extracted information is normalized based on the normalized rule. After the IE Module processes the collected text, the Post-Processing Module first automatically constructs the regulatory networks based on the extracted information and specific existing knowledge, such as gene functional classes' knowledge and subsystem knowledge. Then the Post-Processing Module saves potential informative sentences into the database. Experts can browse these potential sentences and may create new kinds of rules for later use.

Online Tools Subsystem consists of three layers. The top layer of the Online Tools Subsystem is Web-based User Navigation Interface (WUNI), including three parts: web-based browsing, web-based querying engine, and the online building or editing regulatory networks module. The user can browse, build, or edit regulatory networks. The user can also query the functional role, the regulatory relationship between genes, or the reference information. The query engine links various gene information, relation information, reference information, and regulatory network information together and returns the detailed information in a HTML page as well as a visualized graphical result to the user. The middle layer of the Online Tools Subsystem is WWW service layer, including several functional modules, such as the visualization module

(VM), the query analysis module (QA), and the network construction module (NC). These functional modules process the user requests from WUNI, communicate with the bottom layer local database, and return the results to WUNI. At the bottom layer of the Online Tools Subsystem, GRNS implements a relational database system -- the Local Database.

The Analysis Tools Subsystem includes the frequent pattern mining tool and the gene relation prediction tool. GRNS uses an export module to export regulatory networks in Regulatory Networks DB to a standard format. The frequent pattern mining tool reads the exported regulatory networks and uses an inexact match graph mining schema to find the frequent patterns. The frequent pattern mining tool is a standalone tool. It can mine not only frequent patterns in the Regulatory Networks DB but also frequent patterns in other external databases, for example KEGG (Kyoto Encyclopedia of Genes and Genomes)[3]. The gene relation prediction tool uses the prediction algorithm to predict the potential gene relation information based on the current gene relation information in the database.

Data Modeling of GRNS

One objective of GRNS is to automatically extract the gene regulatory information from a collection of unstructured biomedical text. Here, the biomedical text is any research paper. Usually these papers are downloaded from the PubMed database. But what is the definition of the gene regulatory information, and what kind of data are we interested in the information extracting? Basically, GRNS extracts five kinds of data after processing the biomedical text: the gene regulatory relation information, the strain number, the genotype, the phenotype, and unrecognized sentences. A detailed explanation is shown in Table 2-1.

[3] www.kegg.com

Why we need these five kinds of data? Clearly, to construct a gene regulatory network, we need the gene regulatory relation information. The regulation can be at the transcriptional level (activation or repression), the posttranscriptional (mRNA stability) level, the translational level or the post-translational (protein-protein interaction/modification) level. We need the strain number, the genotype, and the phenotype information for the data reliability reason. All this information is part of the evidence of gene regulation. They help the researcher to validate the gene regulatory relation information. For unrecognized sentences, it helps us to discover new rules in the information extraction module.

Visualization Modeling of Gene Regulatory Networks

GRNS automatically provides the visualization of regulatory networks. To do this, we provide visualization modeling of regulatory networks. Visualized regulatory networks include two kinds of information, entities and relations. In GRNS, there are seven kinds of entities: genes, proteins, operons, products, merged-genes, subsystem, and step. An operon is a group of key nucleotides sequences that are controlled and usually function as a unit. GRNS has simple operons and complex operons. In a simple operon, there is no relationship between operon's genes. In a complex operon, there are relationships between the operon's genes. Merged-genes are not natural genes and they are used to improve the layout. Step is used to describe the biomedical process. A subsystem usually means a group of related functional roles which are jointly involved in a specific aspect of the cellular machinery. In GRNS, there are more than ten kinds of relations between entities, such as DNA binding, RNA binding, protein binding, the two-component regulatory system, the signal molecule production, signal sensing, the product, the signal/molecule binding, activate, required for and repress. We provide notations and symbols for visualization modeling of the regulatory networks to generate interactive graphical regulatory networks for subsystems or the whole genome. These notations and symbols are

shown in Figure 2-2. It includes three kinds of information: entities, relations, and color information for different function classes.

Pseudomonas aeruginosa Genome

Pseudomonas aeruginosa is an environmental bacterium, which causes serious human infections, especially in those with reduced immunity, patients with Cystic Fibrosis or severe burns (Larbig, *et al*., 2002; Dasgupta, *et al*., 2001; Rossolini, *et al*., 2005; Shiwani, *et al*., 1997). Gram stain of *Pseudomonas aeruginosa* cells (Todar's Online Textbook of Bacteriology-www.textbookofbacteriology.net) is provided in Figure 2-3. *Pseudomonas aeruginosa* Scanning electron micrograph cells (Todar's Online Textbook of Bacteriology-www.textbookofbacteriology.net) is shown in Figure 2-4. A complicated regulatory network coordinates the expression of various virulence genes as well as different functional groups of genes for an efficient host infection and survival in hostile host environments. Prolonged treatments with antibiotics often result in multi-drug resistant isolates, which eventually cause death in the infected individuals.

A large number of virulence genes and regulatory genes encoded by this organism make this bacterium one of the most successful pathogens on earth. The whole genome sequence of this microorganism was completed several years ago and is freely available to the public. The complete sequence of the genome was the largest bacterial genome sequences data, when published, with 6.3-Mbp in size and encoding 5570 predicted genes. Only 480 encoded protein functions have been demonstrated experimentally, while the rest, including 1059 strongly homologous genes functions, demonstrated experimentally in other organisms, 1524 genes functions proposed based on the presence of a conserved amino acid motif, structural feature or limited homology, and 2507 genes functions homolog of previously reported genes of unknown function, or no homology to any previously reported sequences (Stover, *et al*., 2000). Consistent

with the observed bacterial adaptability to various growth environments through alteration of the gene expression pattern, as high as 8% of the genome encodes transcriptional regulators,

In GRNS, we choose *Pseudomonas aeruginosa* genome as our experimental dataset for two reasons: first, there is an urgent need to develop new antimicrobial strategies for an effective control of this deadly bacterium; second, the complete sequence of the genome was one of the largest bacterial genome sequences data. The GRNS system use genome date independent methods, which makes it easy to apply the GRNS to the other genome by importing other genome related names to the local database.

Table 2-1. Explanation of gene regulatory data.

Name	Explanation	Example
Gene regulation relation	Regulation at different level, including Active, Product, Modification, Location, Repress, DNA/RNA Binding, Two Component, Signal/Molecular Binding, Signal Sensing, Required, Signal Molecular Production, Protein Binding, Steps	Fur represses pchR
Strain number	A strain is a genetic variant or subtype of a virus or bacterium. The strain-number is a number to identify the strain.	DH54
Genotype	The genotype is the specific genetic makeup (the specific genome), in the form of DNA	IA614 *fptA*::VTc (FptA-)
Phenotype	The phenotype of an individual organism is either its total physical appearance and constitution or a specific manifestation of a trait	Elimination of FptA in IA614 yielding strain DH54 restores fptA expression to the level seen in DH143; The loss of FptA in DH54 actually alleviated the apparent repression of pchR expression observed in IA614.
Unrecognized sentences	The sentences include gene name but can not be recognized as including gene regulation information.	One potential product , dihydroaeruginoate (Dha) , was identified in culture supernatants of iron-limited *P. aeruginosa* cells .

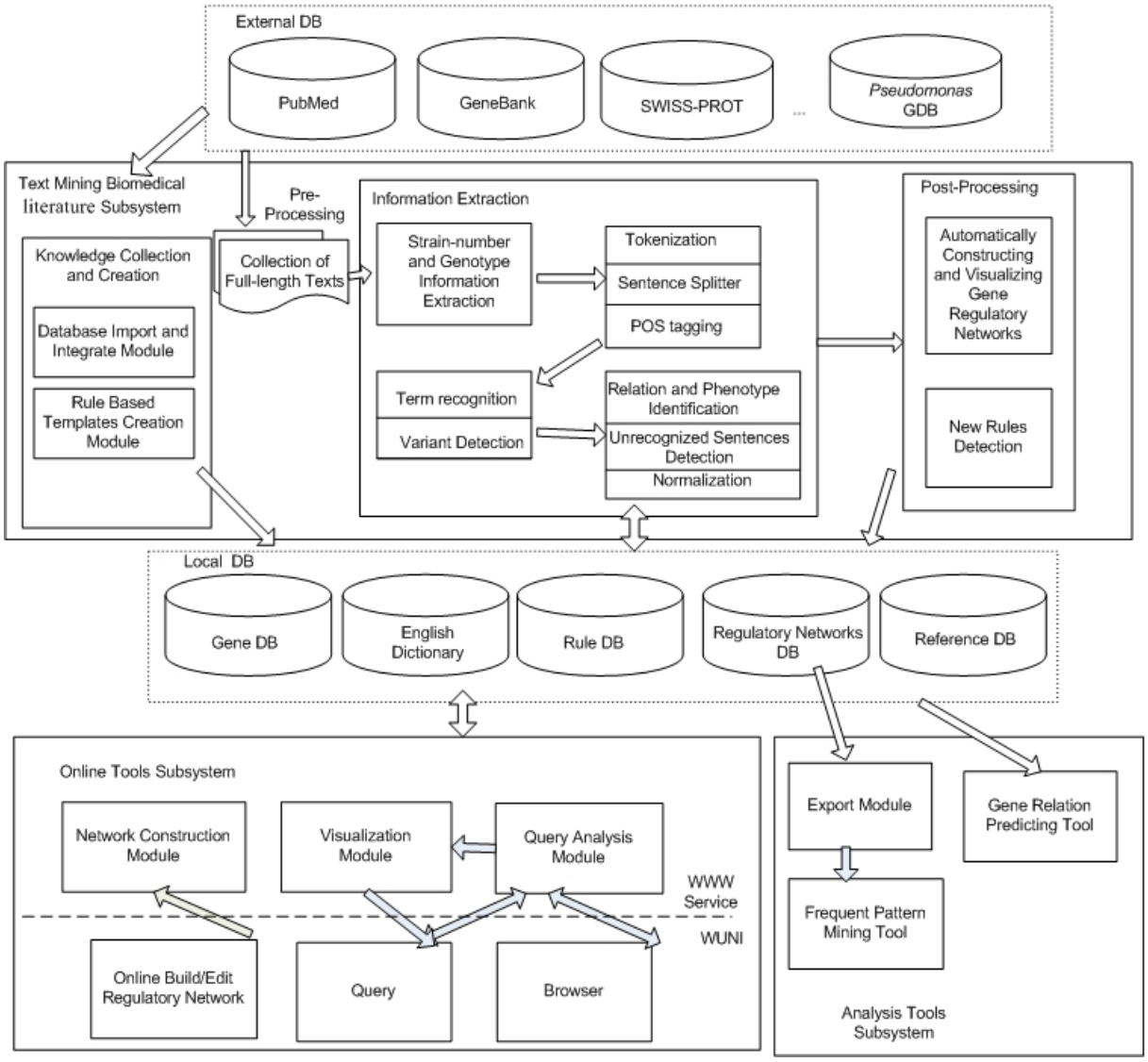


Figure 2-1. System architecture of GRNS.

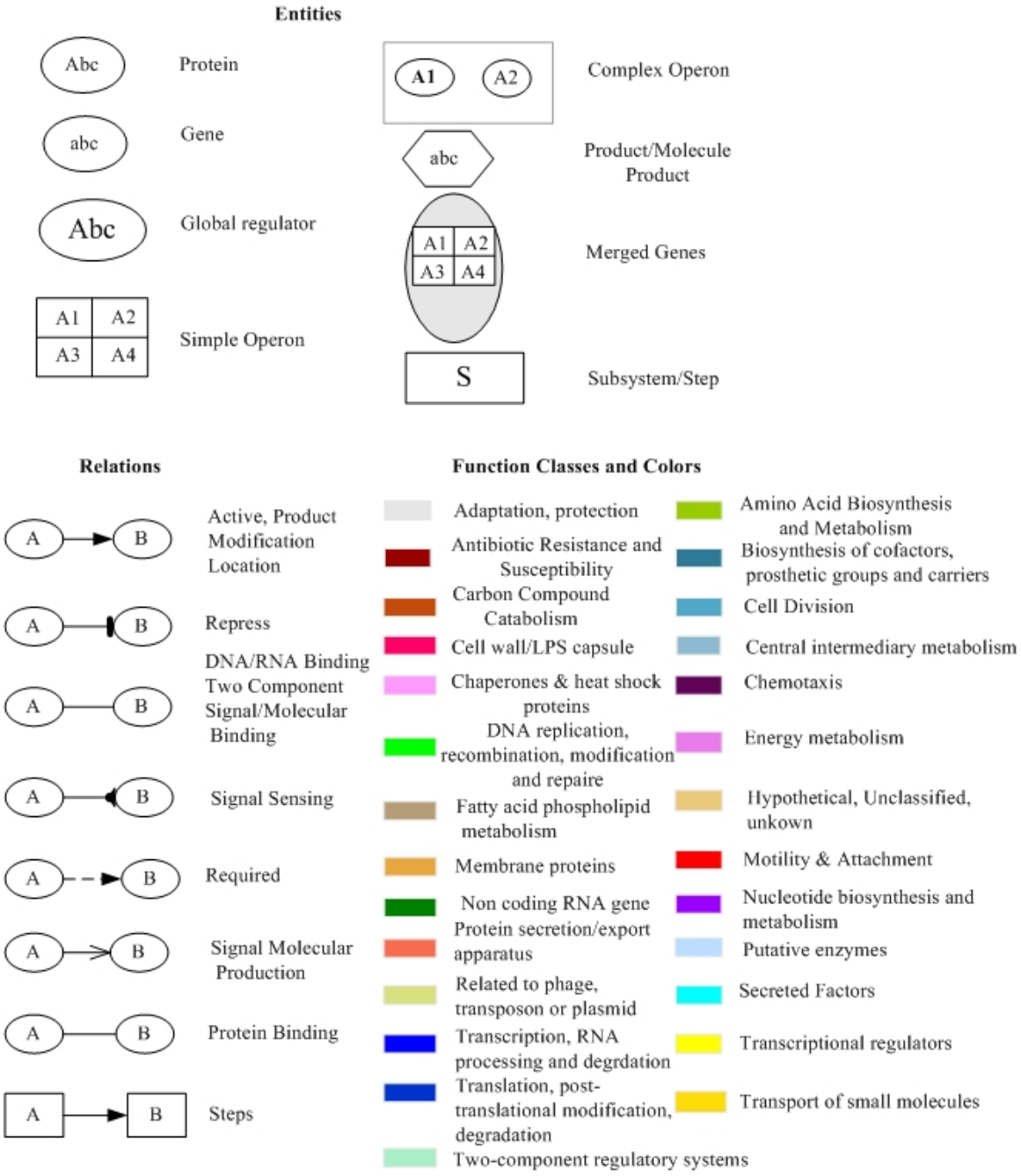


Figure 2-2. Visualization annotation and symbols for GRNS.

Figure 2-3. Gram stain of *Pseudomonas aeruginosa* cells.

Figure 2-4. *Pseudomonas aeruginosa* scanning electron micrograph.

CHAPTER 3

TEXT MINING BIOMEDICAL LITERATURE SUBSYSTEM

In this chapter, we introduce the design and implementation of Text Mining Biomedical Literature Subsystem in GRNS. First, we present text mining biomedical literature's problem definition. Then we provide the design of the Information Extraction Module of the Text Mining Biomedical Literature Subsystem. Next, we show the running results of the Text Mining Biomedical Literature Subsystem for one biomedical paper. Then, we supply the evaluation results of the Text Mining Biomedical Literature Subsystem. In the end, we explain the research contribution of the Text Mining Biomedical Literature Subsystem.

Problem Definition

Text mining technology is used in GRNS to extract the gene regulatory information from the collection of biomedical texts. Text mining is defined as the nontrivial extraction of previously unknown and potentially useful information from texts (Chakrabarti, 2000; Frawley *et al*., 1992). In GRNS, we use the text mining technology to extract gene regulatory information from the collection of biomedical literature. One simple process is shown in Figure 3-1. The problem of text mining biomedical literature in GRNS is defined as given a collection of unstructured biomedical documents, using the text mining technology to extract and store the gene regulatory information to the database. The input, process and output of this problem are shown as follows:

Input: Biomedical document collections. Most biomedical documents are downloaded from the PubMed database.

Process: Using text mining technology to compile and analyze the collection of the document. The text mining technology includes natural language processing and information extraction.

Output: Structured gene regulatory information database entries. The gene regulatory information includes strain number, the genotype, the phenotype, the gene regulation relation information and unrecognized sentences.

Information Extraction Module

The Information Extraction Module (IE) is the most important module in the Text Mining Biomedical Literature Subsystem. A complete text mining process in IE is shown in Figure 3-2. To clearly explain the Information Extraction Process, we use the sentence "FleQ and RpoN positively regulate expression of flhA, fliE, fliL, and fleSR genes, among others." (Jyot *et al*, 2002) as our running example in every step.

Lexicon Construction

Before starting the text mining process, we must build the lexicon. A lexicon is a database of the vocabulary of a particular domain or a language. Usually, there are two kinds lexicon: grammar lexicon and semantic lexicon. We build our lexicon as follows: for every word, the lexicon provides the grammar or semantic part-of-speech tag. We build our grammar lexicon from Eric Brill's tagger program (Brill, 1995). This tagger program can recognize English words and numbers and assign the Part-of-Speech tag to these words. Since our experimental data is the *Pseudomonas aeruginosa* Genome, we use the *Pseudomonas aeruginosa* Genome Database (http://www.pseudomonas.com/) as our main supplement for the semantic lexicon. Besides, we also add some common biomedical terms (such as gene, operon, binding, active, repress and so on) in the semantic lexicon. In case of text mining other genomes in the future, we may add supplemental genome information in the semantic lexicon. Some examples of thegrammar lexicon and the semantic lexicon are shown in Table 3-1.

Strain Table Analysis

The strain number and genotype are important gene regulatory information. Usually, there is a table in the biomedical paper showing all the strains used in the study and their genotypes. We analyze the strain table and extract the genotype and strain number from the table. One difficulty in analyzing the strain table is that we do not have a structured table data directly. We need to extract the table information from the unstructured text. Usually, we do not know where the table starts, when one table column ends, when one table row ends, and when the table ends.

Fortunately, most biomedical papers provide the strain table in a standard format. For example, most of the strain table names include "strains" or "plasmids", such as, "Bacterial strains and plasmids used in this study", "Strains, plasmids, and primers used in this study", "Bacterial strains and plasmids". If we get one line text like "Table …strains … (plasmids) …", usually it means a strain table. Next, we need to recognize how many columns in this table and what the meaning of each column is. In most cases, these columns are also in a standard format. Most tables include three columns, the first one for the strain number, the second one for the genotype, and the last one for the reference information. To recognize these columns, we can follow these patterns. The column name for the strain number usually includes strain or plasmid, such as, "Strain or plasmid ", " Strain, plasmid, or oligonucleotide ". The column name for genotype usually includes genotype or description or characteristics, such as, "Description or sequence", " Characteristics "or "Description". The column name for reference usually includes the reference or the source, such as, "Reference", " Source or Reference". Based on this information, we can recognize the column names. The next step is to recognize the strain number and genotype information. This is a difficult task because we do not have a structured text to separate every column. But we find out that there are some rules to help us to recognize the strain number and genotype information. To recognize the strain number, we find out that most

strain numbers are one word. If the strain number is more than one word, usually the first word of the multi-words strain number is also a strain number itself. One example of multi-words strain number is "PAO1 ncr". Here PAO1 itself is a strain number. We can find the strain number in this way: if the first word is not a strain number recognized before, then the first word is a strain number; otherwise, the first and next word is the strain number. For the genotype, we do not have a very strong rule to recognize the genotype, but we can recognize the genotype by recognizing the reference information. Most the reference information is in this format: "this study" or a number or people name abbreviation. We recognize the reference information first. Once we find where the reference information and strain number are, all other information in this line is genotype information. In the end, we need know where the table ends. Usually, when one table ends, there is one empty line. However, the table does not end after every empty line. If we read one empty line, then we need to recognize what the next line is after the empty line. If the next line is the strain number, the genotype, and the reference, then we continually read the table; otherwise, the table ends.

Tokenization and Sentence Splitter

This process transfers indivisible characters into words and sentences. The text is transformed in one sentence per line format. The output of Tokenization and Sentence Splitter for our running example is shown as follows:

FleQ and RpoN positively regulate expression of flhA , fliE , fliL , and fleSR genes , among others .

Part-of-speech Tagging

Part-of-speech tagging (POS tagging or POST), also called grammatical tagging, is the process of marking up the words in a text corresponding to a particular part of speech, based on both its definition, as well as its context, that is, the relationship with adjacent and related words

in a phrase, a sentence, or a paragraph. A simplified form of this is in the identification of words as nouns, verbs, adjectives, adverbs, etc. To tag the words with POS labels, we use the Brill part-of-speech tagger(Brill, 1995).The output of POS Tagging for our running example is shown as follows:

FleQ/NNP and/CC RpoN/NNP positively/RB regulate/VB expression/NN of/IN flhA/NN ,/, fliE/NN ,/, fliL/NN ,/, and/CC fleSR/NN genes/NNS ,/, among/IN others/NNS ./.

Term Recognition and Variant Detection

After POS tagging, IE performs term recognition and variant detection to recognize discriminating words and Gene/Protein names. Discriminating words recognition is relatively simple: given the pre-defined words set, with the detection of synonym and different verbal form, we can detect the synonym with the help of a synonym dictionary and label the discriminating words with any verbal form. Recognizing the Gene/Protein names is challenging. We need to take care of the problems of the expanded form of abbreviation, homology and aliases. To deal with these problems, we use a gene-dictionary of aliases and abbreviation. We construct the gene-dictionary by combining multi-database recourses, such as the SWISS_PROT and the *Pseudomonas aeruginosa* Genome Database. The output of term recognition and variant detection for our running example shows as follows:

<s>

FleQ nnpg

and CC

RpoN nnpg

positively RB

regulate regv

expression expr

of IN

flhA nnpg

, ,

fliE nnpg

, ,

fliL nnpg

, ,

and CC

fleSR nnpg

genes NNS

, ,

among IN

others NNS

. .

</s>

Relation and Phenotype Identification

We use the cascaded finite state automata to recognize the gene regulatory relation and phenotype information. The cascaded finite state automata are implemented by a CASS parser (Abney, 1996). CASS parser is a robust and speedy partial parser. Our rules for gene regulatory relation and phenotype information recognition are written in the CASS grammar. Some previous systems also use the CASS parser to recognize the regulatory gene/protein relation information. The most famous one is the STRING-IE system in EMBL project (Saric *et al.*, 2005). We follow some basic grammar in STRING-IE with some supplemental grammar rules. For example, there are some examples of our CASS grammar in finding the gene regulatory

relations shown in Tables 3-2, 3-3, and 3-4. These examples can-not be recognized by the STRING-IE CASS grammar. The first line of these tables is the grammar we provide for the CASS parser. The next line is the meaning of this grammar. Then we provide a real sentence from a biomedical paper, the parse tree after being parsed by the CASS parser. In the end, we provide the extracted gene regulatory relation information.

Unrecognized Sentences Detection

It is difficult to build up complete rules to recognize all entities and relations. IE stores the potential sentences to detect possible template candidates. Experts can create new templates based on template candidates' information. We choose the template candidates in this way: if a sentence includes gene/protein names and fails to match the existing pattern, we assign this sentence as a template candidate.

Normalization

The relation and all the entities are normalized based on normalization rules. Some normalization examples are shown in Table 3-5.

Automated Construction and Visualization of Regulatory Networks

After the IE Module, the Text Mining Biomedical Literature Subsystem constructs the regulatory networks based on the entity and relation information extracted from biomedical literature. GRNS provides an automatically interactive visualization method to visualize and integrate the biomedical evidence to the visualized regulatory networks. Interactive visualization of regulatory networks provides an interactive way to browse the regulatory networks.

Results

We show the text mining results for the paper--" Biosynthesis of Pyochelin and Dihydroaeruginoic Acid Requires the Iron-Regulated pchDCBA Operon in *Pseudomonas aeruginosa* " in the Text Mining Biomedical Literature Subsystem (PMID: 8982005) (Serino ***et.***

al, 1997). The results include strain number and genotype information in Figure 3-3. The gene regulatory relation information is shown in Figure3-4. The phenotype information is shown in Figure 3-5. The unrecognized sentences are shown in Figure 3-6. The visualized gene regulatory network based on the extracted gene relation information is shown in Figure 3-7.

Evaluation Results

Currently, thousands of genes and relationship data about *P. aeruginosa* have been collected from more than 200 full-length papers (Greenberg, 2000; Mattick, 2000; Kanehisa, 2000;Wu *et al*., 2005; Wu *et al.*, 2006). We use precision and recall to evaluate the results of the Text Mining Biomedical Literature Subsystem. Precision and recall are the most common parameters when evaluating the IE system. Precision is to evaluate whether the system can only extract correct information, recall is to evaluate whether the system can recognize all useful information (Shatkay *et al*, 2003). The definition of precision and recall is shown as follows:

Precision = number of correctly extracted entities / number of total extracted entities

Recall = number of correctly extracted entities / number of all correct entities

To evaluate the precision and recall of the extracted information, it is necessary to manually analyze all information in the corpus and compare them with the extracted information. There is no automatic tool that can differentiate the correctly extracted entities from unrecognized but correct entities. We randomly select 20 papers from our current corpus and provide the evaluation results in Table 3-6. From the evaluation results in Table 3-6, we can see that the Text Mining Biomedical Literature have high precision and good recall in extracting relation information and other gene regulatory information.

Comparison with Other Biomedical Text Mining System

In this section, we show the comparison result in our text mining subsystem with other biomedical systems.

STRING-IE

The STRING-IE system in EMBL project (Saric *et al.*, 2005) is a rule-based text mining system. It provides a text mining method to extract the gene relation information from PubMed database. One example of the STRING-IE result set is shown in Figure 3-8. Comparing with STRING-IE, our system has several advantages:

4. Our system provides the automatic visualization results for the text mining results. The STRING-IE system provides the user the parse tree, which is not easy for the user to understand. Our system can automatically construct and visualize the gene regulatory networks, which are much easier for user to understand.

5. Our system provides more complete rules for the biomedical text mining. We supplement some rules, which are not provided by STRING-IE. For example, the grammar extracts the relations that "gene regulates itself" expression.

6. Our system provides the sentence evidence information for the extracted gene regulatory relations. So when the user gets the result, he/she also gets why we get the result. The STRING-IE system does not provide the evidence information.

RLIMS-P

RLIMS-P is implemented by Yuan et. al. (Yuan *et. al.*, 2006). It is a rule based approach for extracting information from biomedical text. One example of the RLIMS-P result page is shown in Figure 3-9. Comparing with RLIMS-P, our system has several advantages:

1. Our system can extract the gene regulatory relation for active, repress, product and so on. The RLIMS-P system can only extract Protein Phosphorylation information. So the RLIMS-P is limited comparing with our system.

2. Our system provides the automatic visualization results for the text mining results. The RLIMS-P system only provides the webpage information for the extracted information.

Research Contribution in Text Mining Biomedical Literature Subsystem

There are a lot of similar text mining systems that can extract gene relation information from biomedical literature, such as STRING-IE system in EMBL project (Saric *et al.*, 2005), Arizona Relation Parser System (McDonald *et. al.*, 2004). Compared to other systems, our text mining biomedical literature subsystem mainly has three advantages.

First, the text mining biomedical literature subsystem can extract the strain number and genotype, and the phenotype information from the full-length text. Most text mining systems can extract information from the collection of abstracts. Full-length texts analysis is more complicated and difficult. Moreover, extracting useful information from tables is challenging since most of the biomedical articles are unstructured texts. In our system, we can not only mine full-length text but also extract the strain number and the genotype information from the strain table, and extract the phenotype information from the full-length text. The strain number and genotype, and the phenotype information are good supplemental evidence for the gene regulatory relation information.

Second, we extend the CASS Grammar provided by STRING-IE system (Saric *et al.*, 2005). STRING-IE system offers an organized and useful CASS grammar for biomedical literature mining. However, some important gene relation patterns can not be extracted using the STRING-IE grammar. We supplement some new CASS grammars to help our system recognize more relation information.

Third, most systems provide visualized results for extracted gene relations. But they do not integrate all related information or provide all related information in an interactive fashion. In our system, we link gene information, relation information and mutant information (genotype, phenotype) together and put all information in interactively visualized gene regulatory networks. A user can click entities or relations in the gene regulatory network and receive a collection of related gene regulatory information.

Table 3-1. Lexicon examples.

Word	Grammar Tag	Semantic Tag
regulate	RB	regv
expression	NN	NN
flhA	NN	NNPG
fliE	NN	NNPG

Table 3-2. CASS grammar example1.

Grammar	ex_reg -> nxpg (cma? wdt?) (rb)* (cma? neg?) (vx?) (rb)* (VERB) ownexpr
Explanation	One possible gene relation is gene A regulate its own expression.
Example sentence	PchR negatively regulates its own expression
Parse tree	<s> [ex_reg [nxpg [nxpg [nxpg [nnpgx [nnpg PchR]]]]] [neg negatively] [regv regulates] [ownexpr [prps its] [own own] [expr expression]]] [per .] </s>
Action	<PchR> <REPRESS> <PchR>

Table 3-3. CASS grammar example2.

Grammar	ex_regvd -> nxpg comp? vx nxpg regvd;
Explanation	Gene A is regulated by gene B.
Example sentence	Four promoters (flhA, fliE, fliL, and fleSR) that were RpoN and FleQ regulated were selected for this study.
Parse tree	[ex_regvd [nxpg [nxpg [nxpg [nxpg [nnpgx [nnpg flhA]] [cma ,] [nnpgx [nnpg fliE]] [cma ,] [nnpgx [nnpg fliL]] [cma ,] [cc and] [nnpgx [nnpg fleS]]] [cc and] [nxpg [nnpgx [nnpg fleR] [sym)]]]]] [comp that] [vx [bedr were]] [nxpg [nxpg [nxpg [nnpgx [nnpg RpoN]] [cc and] [nnpgx [nnpg FleQ]]]]] [regvd regulated]]
Action	<RpoN> <REGULATED> <flhA>, <RpoN> <REGULATED> <fliE>, <RpoN> <REGULATED> <fliL>, <RpoN> <REGULATED> <fleSR>, <FleQ> <REGULATED> <flhA>, <FleQ> <REGULATED> <fliE>, <FleQ> <REGULATED> <fliL>, <FleQ> <REGULATED> <fleSR>

Table 3-4. CASS grammar example3.

Grammar	locate_gene (vx?) (ADV)* (vx?) (rb)* (VERB) (in | of | by) nxpg;
Explanation	Gene A located in upstream/downstream Gene B Relate gene C.
Example sentence	fimS located immediately upstream of algR is also required for twitching motility
Parse tree	<s> [ex_reg_at [locate_gene [nxpg [nxpg [nnpgx [nnpg fimS]] [vbn located]]] [rb immediately] [upstream upstream] [of of] [nxpg [nxpg [nnpgx [nnpg algR]]]]] [vx [bez is]] [rb also] [requvd required] [in for] [nxpg [nxpg [nxpg [nnpgx [twitching twitching] [motility motility]]]]]] </s>
Action	< twitching motility> <REQUIRE> <fimS>

Table 3-5. Normalization examples.

Terms/Relations	Normalized terms/Relations	Normalized Type
Pseudomonas quinolone signal	PQS	mapping of expanded forms to acronyms
Operon fleSR	fleS, fleR	mapping of operon to the list of genes
Gene A is repressed by Gene B	Gene B repress Gene B	Mapping relation information in stand form
Gene A negatively regulate Gene B	Gene A repress Gene B	Mapping relation name in stand form

Table 3-6. Evaluation result for text mining biomedical literature subsystem.

Name	Precision	Recall
Strain number	0.93	0.92
Genotype	0.90	0.89
Gene regulatory relation	0.91	0.79
Phenotype	0.87	0.74

Figure 3-1. A simple text mining process.

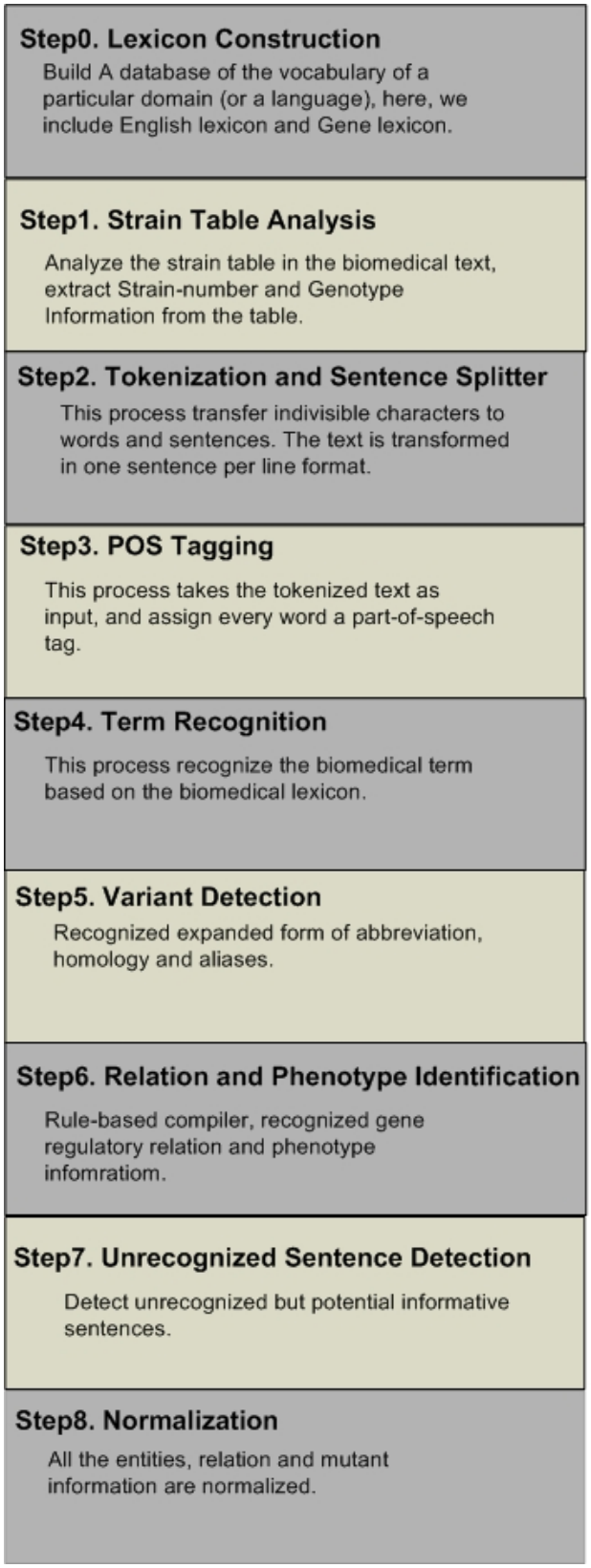


Figure 3-2. A complete text mining process in IE.

Strains Number	Genotype
DH5a	recA1 endA1 hsdR17 deoR thi-1 supE44 gyrA96 relA1 D(lacZYA-argF)U169 (f80dlacZDM15)
PALS128	pvdB
PALS128-17	pvdB pchB or pchA
PAO1 Prototroph	ATCC
PAO307	argC54
PAO4104	pyu-9010 dcu-9008 trpE9051
PAO6285	pchD::v
PAO6286	pvdB pchD::VSm/Sp
PAO951	proA137
PAOA4	fur-4
pBluescript	I KS-Cloning vector; ColE1 replicon; Apr
pKT240	IncQ Mob Apr/Cbr Kmr
pME3087	Suicide vector; Tcr; ColE1 replicon
pME3300	pLAFR3 carrying a 28-kb BamHI-HindIII fragment of P. aeruginosa PAO1 DNA, containing the structural genes for salicylate and pyochelin biosynthesis
pME3315	pQF10 carrying a 6.7-kb BglII-KpnI fragment containing the pchDCBA genes
pME3318	pQF10 carrying a 4.6-kb EcoRI-SalI fragment containing the pchDCBA genes
pME3349	pBluescript carrying a 953-bp XmaI-EcoRI fragment (EcoRI site from Tn1725 insertion 9)
pME6122	pKT240 carrying a 3.7-kb EcoRI-DraI fragment containing pchD9-9lacZ derived from pNM480; Cbr/Apr
pNM480	Apr; ColE1 replicon, 9lacZY1
pQF10	Broad-host-range vector, ColE1-pRO1600 replicon; Apr Tcr
S17-1	thi pro hsdR recA; chromosomal RP4 (Tra1 Tcs Kms Aps)
XL1-Blue	recA1 lac endA1 gyrA96 thi hsdR17 supE44 relA1/F9 (proA1B1 lacIq lacZDM15 Tn10)

Figure 3-3. Strain number and genotype information extracted from paper "Biosynthesis of pyochelin and dihydroaeruginoic acid requires the iron-regulated pchDCBA operon in *Pseudomonas aeruginosa*".

Gene	Relation	Gene	Sentence
iron	REPRESS	salicylate	Excess iron represses synthesis of pyoverdin , salicylate , and pyochelin in P. aeruginosa
iron	REPRESS	pyoverdin	Excess iron represses synthesis of pyoverdin , salicylate , and pyochelin in P. aeruginosa
iron	REPRESS	pyochelin	Excess iron represses synthesis of pyoverdin , salicylate , and pyochelin in P. aeruginosa
Fur	REPRESS	pchR	Fur represses transcription of pchR
iron	REPRESS	FptA	iron represses FptA
pyochelin	REQUIRE	PchR	PchR is required for synthesis of pyochelin and pyochelin
PchR	REPRESS	PchR	PchR negatively regulates its own expression
PvdS	REGULATE	pyochelin	PvdS , putative sigma factor , and PchR , member of AraC family of transcriptional activators , are positive regulators of pyoverdin and pyochelin synthesis
PchR	REGULATE	pyochelin	PvdS , putative sigma factor , and PchR , member of AraC family of transcriptional activators , are positive regulators of pyoverdin and pyochelin synthesis
PchR	REGULATE	pyoverdin	PvdS , putative sigma factor , and PchR , member of AraC family of transcriptional activators , are positive regulators of pyoverdin and pyochelin synthesis
PvdS	REGULATE	pyoverdin	PvdS , putative sigma factor , and PchR , member of AraC family of transcriptional activators , are positive regulators of pyoverdin and pyochelin synthesis
Pyochelin	REQUIRE	pchD	Pyochelin and Dha Requires Iron-Regulated pchD and pchC and pchB and pchA Operon in Pseudomonas aeruginosa
Pyochelin	REQUIRE	pchC	Pyochelin and Dha Requires Iron-Regulated pchD and pchC and pchB and pchA Operon in Pseudomonas aeruginosa
Pyochelin	REQUIRE	pchB	Pyochelin and Dha Requires Iron-Regulated pchD and pchC and pchB and pchA Operon in Pseudomonas aeruginosa
Pyochelin	REQUIRE	pchA	Pyochelin and Dha Requires Iron-Regulated pchD and pchC and pchB and pchA Operon in Pseudomonas aeruginosa
Dha	REQUIRE	pchA	Pyochelin and Dha Requires Iron-Regulated pchD and pchC and pchB and pchA Operon in Pseudomonas aeruginosa

Figure 3-4. Part of gene regulatory relation information extracted from paper "Biosynthesis of pyochelin and dihydroaeruginoic acid requires the iron-regulated pchDCBA operon in *Pseudomonas aeruginosa*".

Strains Number / Gene	Phenotype
PALS128	The pyochelin standard was isolated as described previously (13) from a culture filtrate of strain PALS128 / pME3300 grown in DCAA medium (54) and purified by HPLC .
PALS128-17	The salicylate-and pyochelin-negative mutant PALS128-17 of P. aeruginosa is complemented by plas-Downloaded from jb.asm.org at Univ of Florida on April 25 , 2007 VOL. 179 , 1997 PYOCHELIN BIOSYNTHETIC GENES pchDCBA IN P .
PALS128-17	Three insertions (designated 7 , 8 , and 9 ; Fig. 1 and 2) lying in a 2.5-kb region upstream of pchB abolished complementation of strain PALS128-17 , as did others (insertions 3 , 4 , and 6) in pchB and pchA which were previously described (54) .
PALS128-17	When the pyoverdin- , salicylate- , and pyochelin-negative strain PALS128-17 was complemented for pyochelin production , e.g. , by plasmid pME3315 , an orange halo was formed on CAS agar .
PAO6285	Insertion 7 was used to construct the mutants PAO6285 and PAO6286 .
PAO6285	The best evidence comes from the salicylate-positive , Dha-and pyochelin-negative phenotype of the pchD mutant PAO6285 (Table 2) .
PAO6285	The v insertion mutant PAO6285 did not produce Dha and pyochelin but excreted salicylate , albeit in small amounts (Table 2) .
PAO6286	A transducing lysate of phage E79tv-2 was prepared from strain PAO6286 (pchD : : VSm / Sp) and used to transduce strains PAO4104 (trpE) , PAO307 (argC) , and PAO951 (proA) to prototrophy .
PAO6286	The VSm / Sp element used to construct the pchD mutant PAO6286 (triangle below the chromosome) is shown .
pBluescript	Subclones for DNA sequencing were generated in pBluescript KS2 (Stratagene) by nested deletions using an exonuclease III Erase-a-base kit (Promega) .
pKT240	The pchD9-9lacZ fusion was inserted on a 3.8-kb EcoRI-DraI fragment into the broad-host-range vector pKT240 cut with EcoRI and SmaI , resulting in pME6122 .
pME3087	Both insertion mutations were transferred to the chromosome of the wild-type strain PAO1 (Fig. 1) by marker exchange via the suicide plasmid pME3087 (see Materials and Methods) .
pME3087	The v mutation was transferred on a 1.6-kb BglII-KpnI fragment (Fig. 1) to the ColE1-based suicide vector pME3087 (64) .
pME3300	At least some genes involved in the transformation of compound X into pyochelin are located in the vicinity of the pchR-pchDCBA region on the cosmid pME3300 (54 ; our unpublished results) .
pME3315	A 4.6-kb EcoRI-SalI fragment , obtained from plasmid pME3315 containing transposon insertion 2 , was cloned into vector pQF10 , resulting in plasmid pME3318 .

Figure 3-5. Part of phenotype information extracted from paper "Biosynthesis of pyochelin and dihydroaeruginoic acid requires the iron-regulated pchDCBA operon in *Pseudomonas aeruginosa*".

One potential product , dihydroaeruginoate (Dha) , was identified in culture supernatants of iron-limited P. aeruginosa cells .
antifungal antibiotic Dha is thought to arise from reaction of salicylate with cysteine , followed by cyclization of cysteine .
pchD and pchC and pchB and pchA genes constitute transcriptional unit
full-length pchD and pchC and pchB and pchA transcript of ca. 4.4 kb could be detected in iron-deprived , growing cells of P. aeruginosa .
Pseudomonas aeruginosa , ubiquitous bacterium and opportunistic pathogen , produces three siderophores during growth under iron-limiting conditions : pyoverdin , salicylate , and pyochelin (2 , 11 , 12 , 36 , 60) .

Figure 3-6. Part of unrecognized sentences extracted from paper "Biosynthesis of pyochelin and dihydroaeruginoic acid requires the iron-regulated pchDCBA operon in *Pseudomonas aeruginosa*".

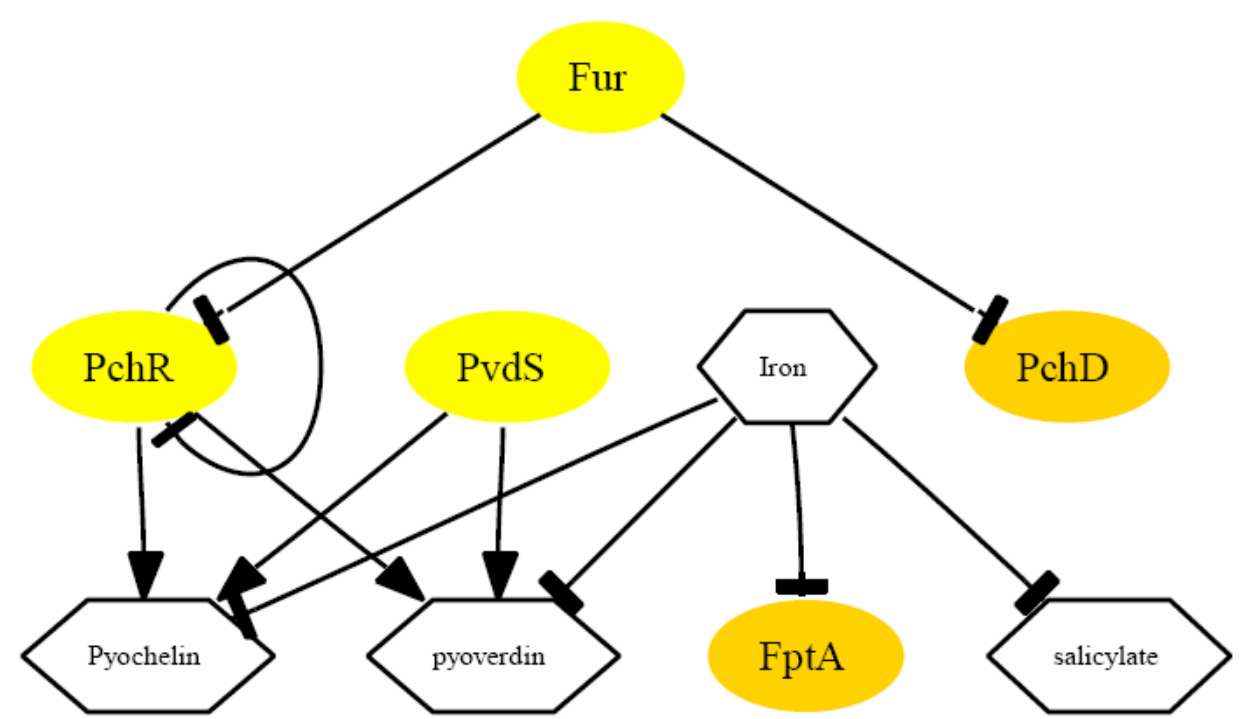


Figure 3-7. Visualized gene regulatory network for paper "Biosynthesis of pyochelin and dihydroaeruginoic acid requires the iron-regulated pchDCBA operon in *Pseudomonas aeruginosa*".

```
|
[s
  [pp We]
  [vvd analyzed]
  [dt the]
  [ax
    [jj resistant]]
  [mut mutants]
  [cc and]
  [showv showed]
  [comp that]
  [nxrest
    [nn Ser-82]]
  [of of]
  [nxpg
    [nxpg
      [nnpgx
        [nnpg GyrA]]]]
  [cma ,]
  [ax
    [jj equivalent]]
  [to to]
  [nxrest
    [nn resistance]
    [jj hot]
    [org spot]
    [cd Ser-83]]
```

Figure 3-8. Part of *E.Coli* result from the STRING-IE system.

PMID Mapping to UniprotKB Advanced Search

Protein AC/ID	Protein Name	Organism Name	PIRSF ID	Matched Fields
P18688/KPB1_RABIT /ProClass UniProtKB/Swiss-Prot	Phosphorylase b kinase regulatory subunit alpha, skeletal muscle isoform (Phosphorylase kinase alpha M subunit)	Oryctolagus cuniculus (Rabbit)	PIRSF015844	PubMed ID=>2108025
P46020/KPB1_HUMAN /ProClass UniProtKB/Swiss-Prot	Phosphorylase b kinase regulatory subunit alpha, skeletal muscle isoform (Phosphorylase kinase alpha M subunit)	Homo sapiens (Human)	PIRSF015844	PubMed ID=>2108025

Name Mapping to UniprotKB

Figure 3-9. One example of result web page from the RLIMS-P system.

CHAPTER 4
ONLINE TOOLS SUBSYSTEM

In this chapter, we introduce the design and implementation of the Online Tools Subsystem in GRNS. First, we present several novel algorithms in the Online Tools Subsystem, and then show several results in the Online Tools Subsystem, including some visualized regulatory networks in GRNS, query results in GRNS, and an online edit/building regulatory network interface.

Novel Algorithms in Online Tools Subsystem

In this section, we introduce several algorithms in the Online Tools Subsystem, including the automated visualization algorithm, the network extraction, and the reconstruction algorithm (NERA) and several algorithms based on NERA.

Automated Visualization Algorithm

In GRNS, one basic part is the automated graph visualization. With the automated graph visualization, we can provide researchers with a clear view on the gene functions or the relationships among several regulatory elements. In GRNS, we use the Graphviz Software (Gansner, *et al.*, 1993,1999) to do the automated graph drawing. Graphviz uses the dot's algorithm, which provides an efficient way to rank the nodes, some heuristic approaches to reduce edge crossings, a scheme for computing the node coordinates as a ranking assignment problem and a method for setting spline control points. Graphviz can automatically convert a dot format text document into a directed graph and provide us the convenience of the automatic graph drawing.

However, the dot's algorithm has some limitations concerning the automatic drawing of the interactive regulatory networks. The dot's algorithm implements an optimal ranking assignment with a heuristic weight function and local transpositions to reduce crossings. One basic idea of

the dot's algorithm is to assign a vertex rank value based on the direct edge. Given the edges, the layout of the graph is decided. That means, when the relationships information in the GRDB is given, the layout for an interactive network is decided. However, to obtain a good layout for the interactive regulatory networks, we need to make some extra effort. First, due to the complexity of the interactive regulatory networks, we need some ways to save the layout space and give the users a clear view. Second, because of some requirements of gene relationships, we need a more specific layout. We need some heuristic methods to improve the layout of interactive regulatory networks. We introduce three heuristic methods for improving the layout in the visualization of the regulatory networks in Figures 4-1, 4-2, and 4-3. Some examples of layout of before and after using these heuristic methods are shown in Figure 4-4. All examples are small portions of some regulatory networks in GRNS.

Network Extraction and Reconstruction Algorithm (NERA)

The network extraction and reconstruction algorithm (NERA) is another important algorithm in GRNS. A lot of other algorithms in GRNS are based on NERA. The function of NERA can be described as follows: if there are a collection of networks, NERA can extract data from these networks and generate a new network based on the collection of data and some specific conditions. One example of NERA is: given a collection of regulatory networks, NERA can generate a new regulatory network, which only includes regulatory relationships and regulatory entities related to one regulatory entity C. The NERA is described as follows.

Algorithm Name: NERA

Input: 1) A collection of regulatory networks R_1, R_2…R_n.

2) Type: genes or relations or regulatory networks or reference

3) Subset, a collection of entities, can be genes or relations or

regulatory networks

Output: new regulatory network N

Entities={};

Relations={};

Operons={};

For every regulatory network R_i

{

For every entity e in R_i

If e is not in Entities and e is related to Subset, add e to Entities

For every operon o in R_i

If o is not in Operons and Operons does not have an operon o'

has same entities with o and o is related Subset, add o to

Operons

For every relation r in R_i

If r is not in Relations and Relations does not have a relation r'

has same meaning with r and r is related to Subset

{ add r to Relations

If entity in r (e_1 or e_2) is an operon which Operons not

contain it

{ find an operon o1 in Operon which contain same

entity with e1 or e2

Mapping e1 or e2 to o1

}

}

}

N= { Entities, Relations, Operons};

Based on NERA, there are several other algorithms useful in network reconstruction and the visualization of query results. The following are two algorithms based on NERA: Merge-Network Algorithm and Linking Algorithm.

Algorithm Name: Merge-Network Algorithm

Input: A collection of regulatory networks R_1, R_2…R_n.

Output: New regulatory network N

When there are several regulatory networks for different subsystems,

generate a regulatory network to represents all these regulatory networks

Given subsystem regulatory networks R_1, R_2…R_n,

Call NERA({R_1, R_2…R_n }, 'network', { R_1, R_2…R_n }}

One example of Merge-Network Algorithm is in shown in Figure 4-5.

Algorithm Name: Linking Algorithm

Input: a PubMed Paper ID

Output: new regulatory network N

When we need visualized the query result about a reference paper.

Subset= Entities in this reference paper Union Relations in this reference paper

Call NERA (all regulatory networks, 'reference', Subset)

Results

Visualized Regulatory Networks

Currently, thousands of genes and relationship data about *P. aeruginosa* have been collected (Greenberg, 2000; Mattick, 2000; Kanehisa, 2000;Wu et al., 2005; Wu et al., 2006). Interactive visualization of regulatory networks for eight important subsystems -- Flagella, Pili, Type III secretion subsystem, Iron acquisition, Quorum sensing, Biofilm, Alginate synthesis and Multi-drug efflux subsystem -- have been constructed. In this section, we give a global view of *P. aeruginosa*, a detailed explanation of the Type III secretion subsystem and the Iron acquisition Subsystem.

Global view of *P. aeruginosa*

The global view for *Pseudomonas aeruginosa* Genome is shown in Figure 4-6. The global network includes eight subsystems: Flagella, Pili, Type III secretion system, Iron acquisition, Quorum sensing, Biofilm, Alginate synthesis, and the Multi-drug efflux system. One subsystem may affect the activation of others. The relationships between different subsystems are presented as red lines in the network. For example, the Flagella and Pili are required for the Biofilm subsystems. *P. aeruginosa* strains with defect in either Flagella or Pili cannot form the Biofilm. Meanwhile, different subsystems are coordinately regulated by proteins. Since those proteins regulate many genes in different subsystems, they are named as global regulators. The

relationships between global regulators and subsystems are presented as black lines in the network. For instance, a global regulator, Vfr, activates the Pili, the Type III secretion system, Iron acquisition, and Quorum sensing subsystems and represses the Flagella subsystem.

Type III secretion subsystem

The regulatory network of Type III secretion subsystem (TTSS for short) is shown in Figure 4-7. The TTSS is an important virulence factor of *P. aeruginosa*: it inhibits host defense systems by inducing apoptosis in macrophages, polymorphonuclear phagocytes, and cells. The TTSS contains a syringe like apparatus, which can directly inject the effector proteins from the bacterium cytoplasm into the host cell cytosol, causing cell death. The *P. aeruginosa* TTSS machinery is encoded by 31 genes arranged in four operons on the chromosome. Four effector proteins, ExoS, ExoT, ExoY and ExoU have been found in *P. aeruginosa*. According to the current working model, the needle forms a pole in the host cell membrane, and the effector proteins are delivered through the hollow needle. Based on published research, we divide the TTSS translocation process into six steps, as presented in the black boxes. Following the boxes, we use the "Type III Secretion System" to represent the overall function of this subsystem. We find that the regulatory network can clearly describe the relationship in the Type III secretion subsystem.

Iron acquisition subsystem

The interactive regulatory network for the Iron acquisitions subsystem is shown in Figure 4-8. The Iron acquisition subsystem acquires iron from surrounding environment. To do this, *P. aeruginosa* produces and secrets an iron-chelating compound, named siderophore. Two types of siderophores: pyoverdine and pyochelin, are produced by *P. aeruginosa*. The pyoverdine and pyochelin synthesis genes and receptors are under the negative control of a regulator-Fur.

Query Results

In this section, we provide an example of the query results for the PubMed paper -- "Characterization of a new efflux pump, MexGHI-OpmD, from *Pseudomonas aeruginosa* that confers resistance to vanadium" (PMID: 12177331), and give an example of the query gene results for ToxR.

The query results for Pubmed Paper --"Characterization of a new efflux pump, MexGHI-OpmD, from *Pseudomonas aeruginosa* that confers resistance to vanadium" -- PMID 12177331(Aendekerk *et al*., 2002) is shown in Figure 4-9. We find that GRNS can summarize all related information in one page, and automatically visualize related regulatory entities and relations. In this way, the researcher can have a clear idea about the query paper. Furthermore, the author can employ GRNS by inputting the relations information about the author's paper. GRNS can integrate the newly discovered data and merge the data into our gene regulatory networks. At the same time, GRNS can generate a visualization of those regulatory networks for the author. When other users query this author's paper, the visualization results can be returned to the query users.

The query results for gene "ToxR" is shown in Figure 4-10. We find that GRNS can summarize all related information in one page, and automatically visualize related regulatory entities and the relationship related to the query gene. In this way, the researcher can have a clear idea about the relation information about the query gene. Besides, since all entities and relation in the visualized result can be interactively clicked, the researcher can get detailed information about the related entities and the relations.

Online Build/Edit Tool for Regulatory Networks

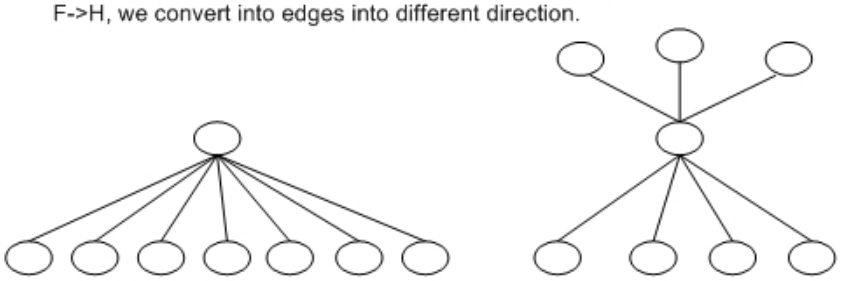
Part of interface for the Online Build/Edit Tool for Regulatory Networks is shown in the Figure 4-11. GRNS provides researchers with a web-based user-friendly tool to build regulatory

networks. Moreover, the automated visualization tool provides the user the visualization of the network once the user completes the data input. The GRNS also provides automatically information linking schema to collect all useful information for the regulatory networks. In this way, the user can take advantage of GRNS without inputting all detail information.

Star-like Transformation

Reason Using This Method:
When there are a lot of edges like A->H,B->H,C->H,D->H,E->H,F->H, all vertices A,B, C,D,E, F will be in same level, which will use a lot of space for this level

Method:
When there are a lot of edges like A->H,B->H,C->H,D->H,E->H, F->H, we convert into edges into different direction.



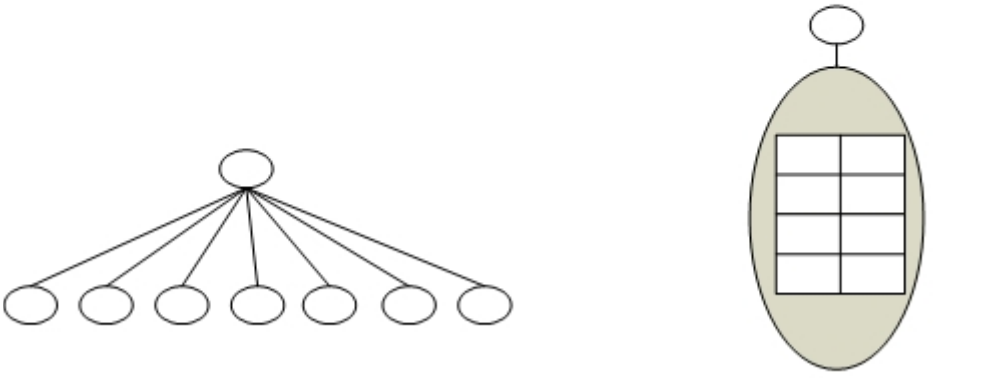
Before start-like transformation After start-like transformation

Figure 4-1. Start-like transformation.

Merge Transformation

Reason Using This Method:
When there are a lot of vertices have the same relationship to other vertices, use merge operation can save a lot of space

Method:
Merge vertices A,B, C,D,E...into one vertex, this new vertex is a complex vertex and include all information about A, B, C,D,E…



Before Merge transformation After Merge transformation

Figure 4-2. Merge transformation.

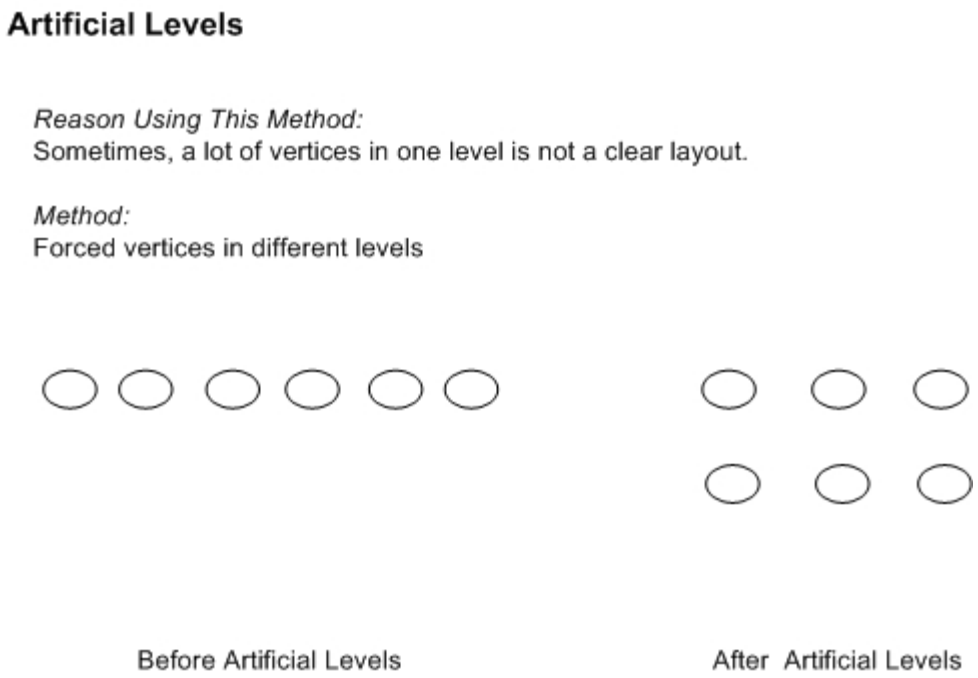


Figure 4-3. Artificial levels.

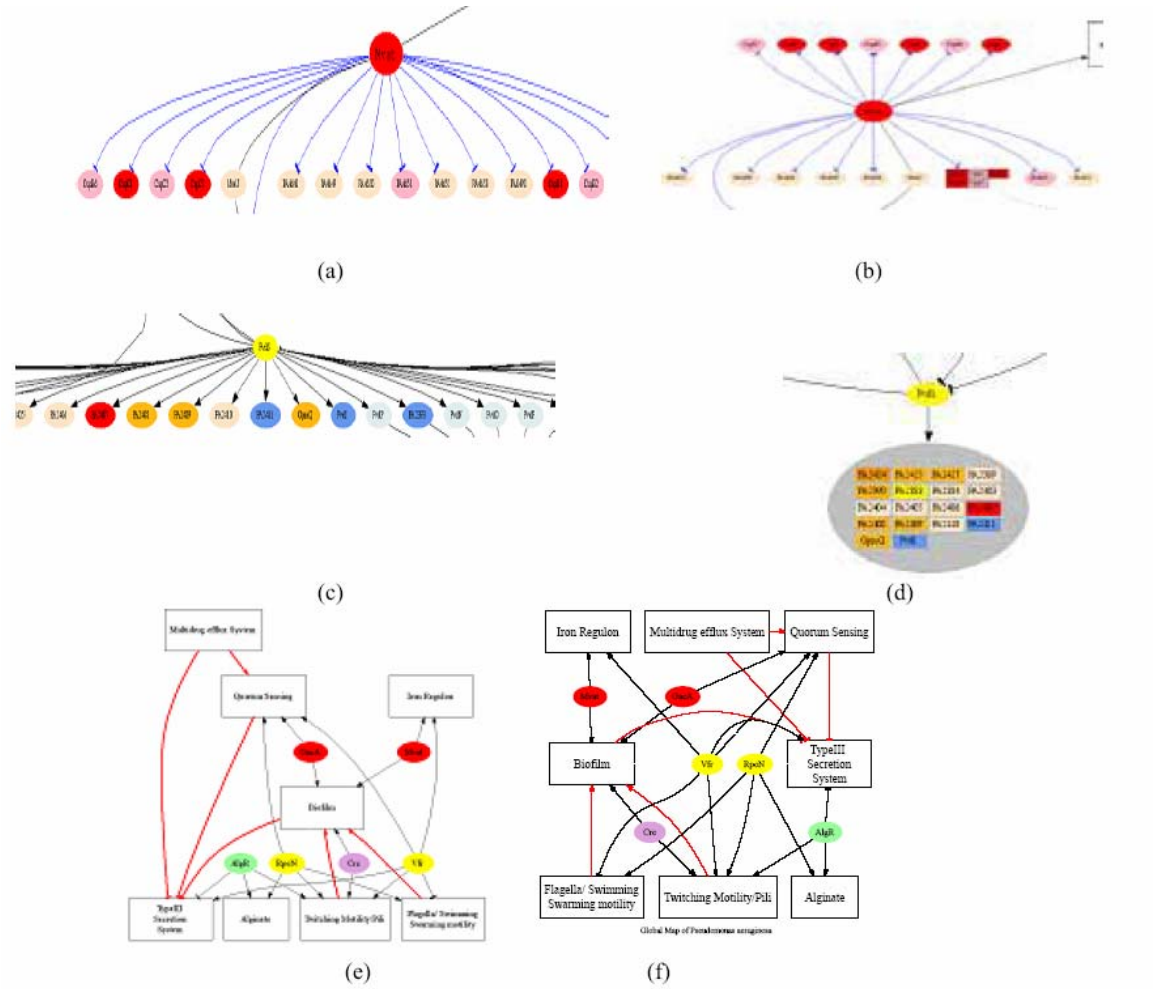


Figure 4-4. Examples before and after using heuristic methods. (a) before star-like transformation (b) after star-like transformation (c) before merge operation (d) after merge operation (e) before artificial levels (f) after artificial levels.

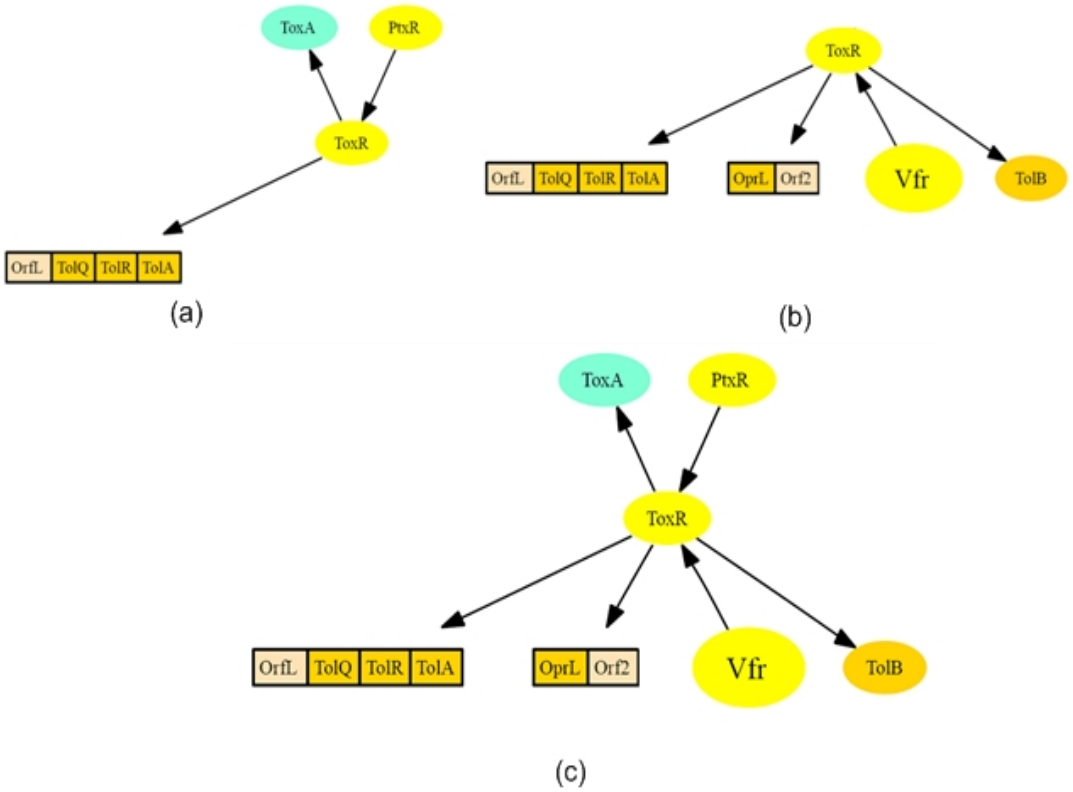


Figure 4-5. One example of merge-network algorithm (a) regulatory network R1. (b) regulatory network R2. (c) new regulatory network N.

.

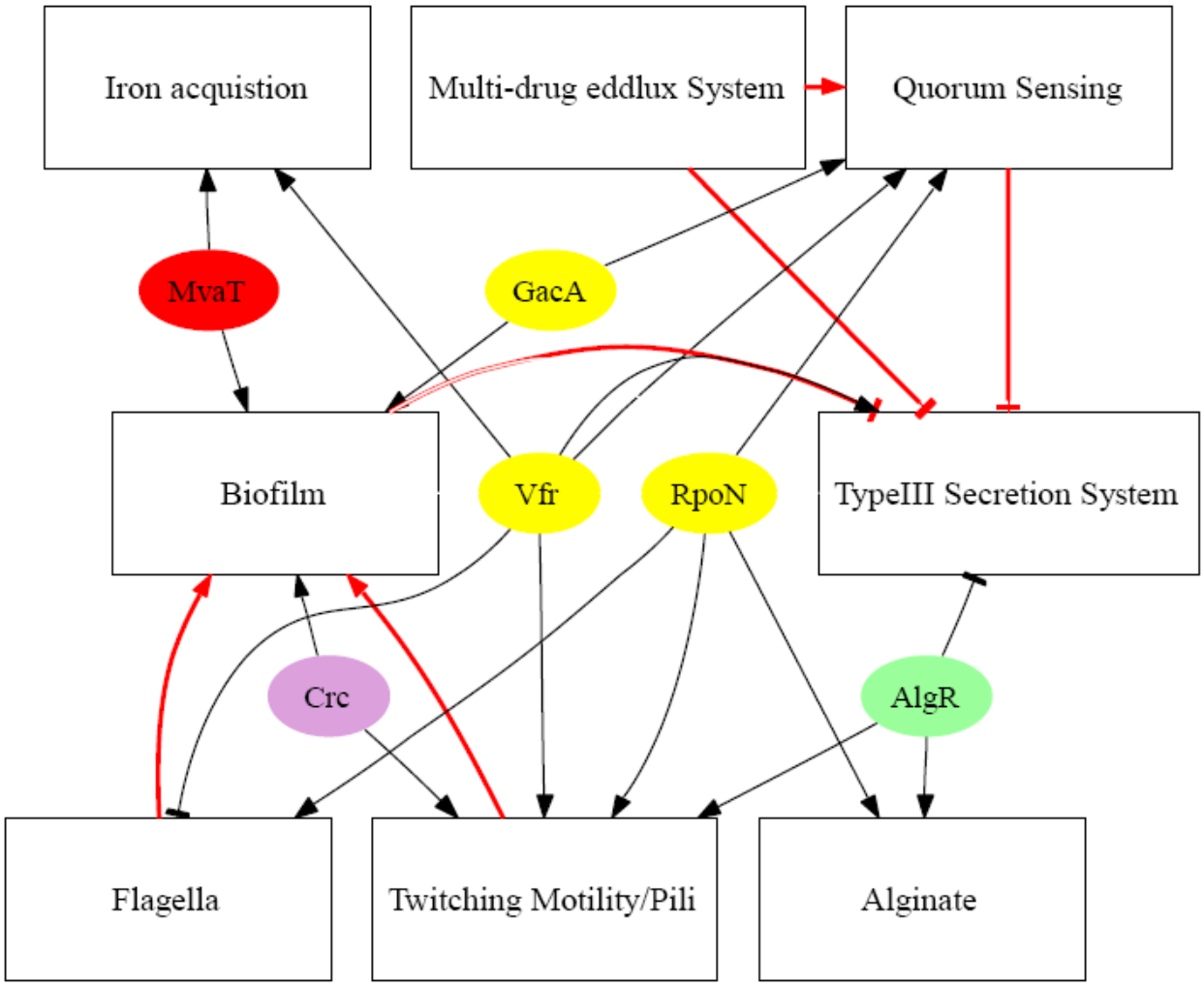


Figure 4-6. Global view of *Pseudomonas aeruginosa*.

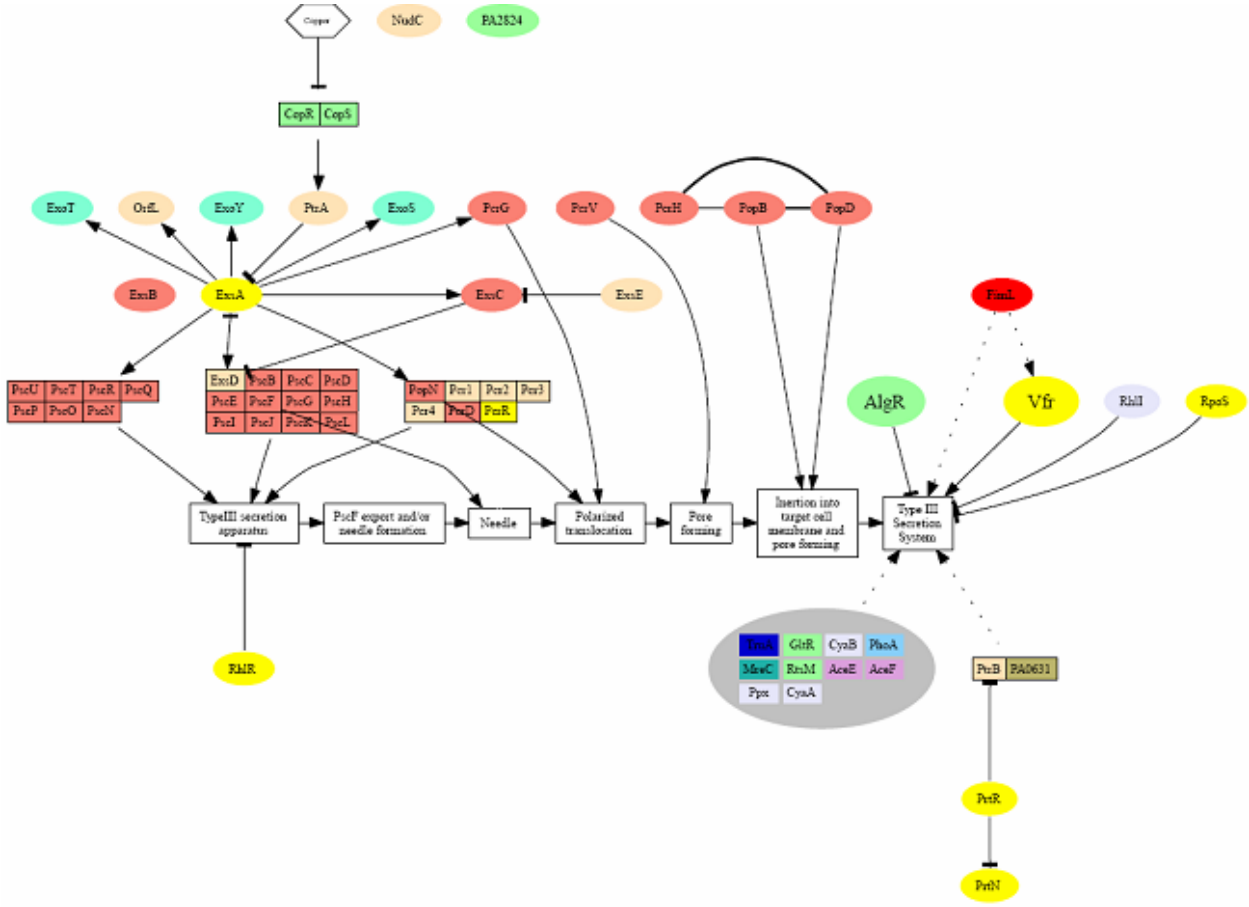


Figure 4-7. Regulatory network of type III secretion subsystem.

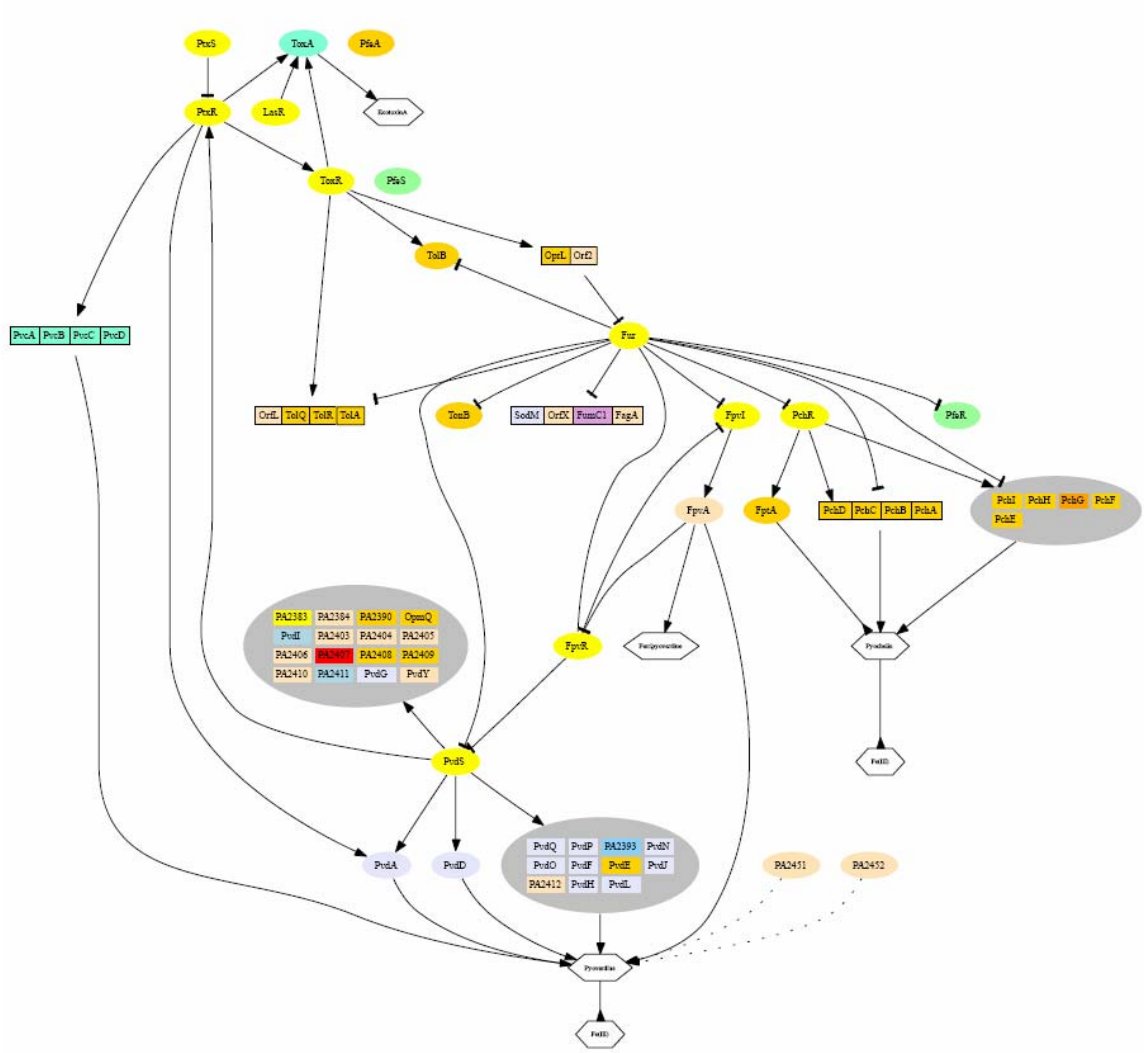


Figure 4-8. Regulatory network of iron acquisition subsystem.

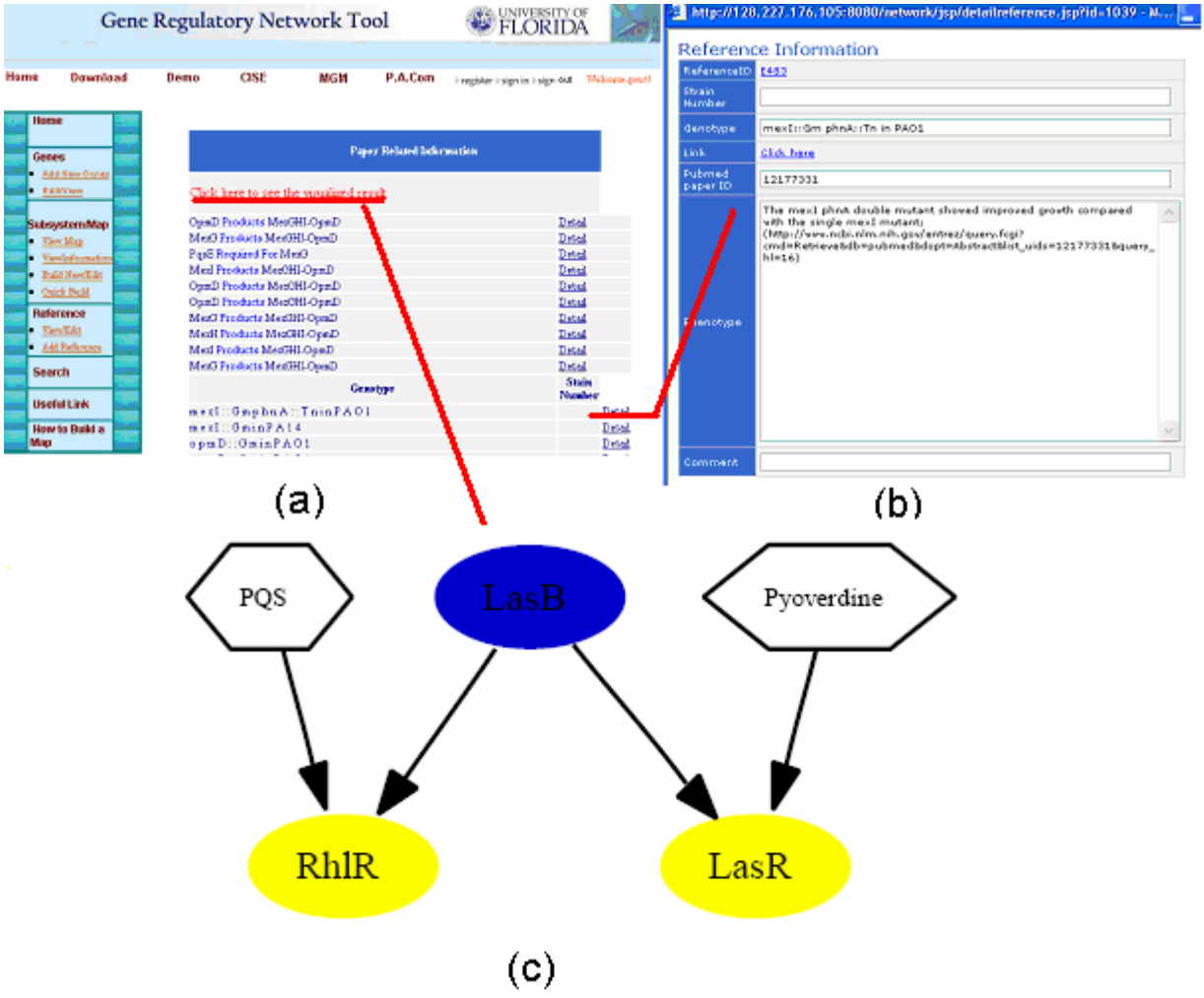


Figure 4-9. Query result for paper PMID 12177331. (a) detailed page for paper PMID 12177331 (b) mutant information for paper PMID 12177331 (c) visualized result for paper PMID 12177331.

PA0707

Click here to see the visualized result

Protein Name	ToxR
Link to www.pseudomonas.com	http://www.pseudomonas.com/AnnotationByPAU.asp?PA=PA0707
Related Subsystems	Global Detail Iron acquisition
Related Operon	
Related Relation	Vfr Actives ToxR
	ToxR Actives Operon [OrfLTolQTolRTolA]
	ToxR Actives TolB
	ToxR Actives Operon[OprLOrf2]
	ToxR Actives ToxA
	PtxR Actives ToxR
	PtxR Actives ToxR
	ToxR Actives ToxA
	ToxR Actives TolB
	Vfr Actives ToxR
	ToxR Actives Operon[OprLOrf2]
	ToxR Actives Operon [OrfLTolQTolRTolA]

(a)

Genotype	Stain Number	
ETA-andprotease-deficientNTG-generatedmutantofPA103	PA103-15PA103-16PA103-19	Detail
ETA-andprotease-deficientNTG-generatedmutantofPA103	PA103-15PA103-16PA103-19	Detail
regABreplacedwithGminPA103		Detail
regABreplacedwithGminPA103		Detail

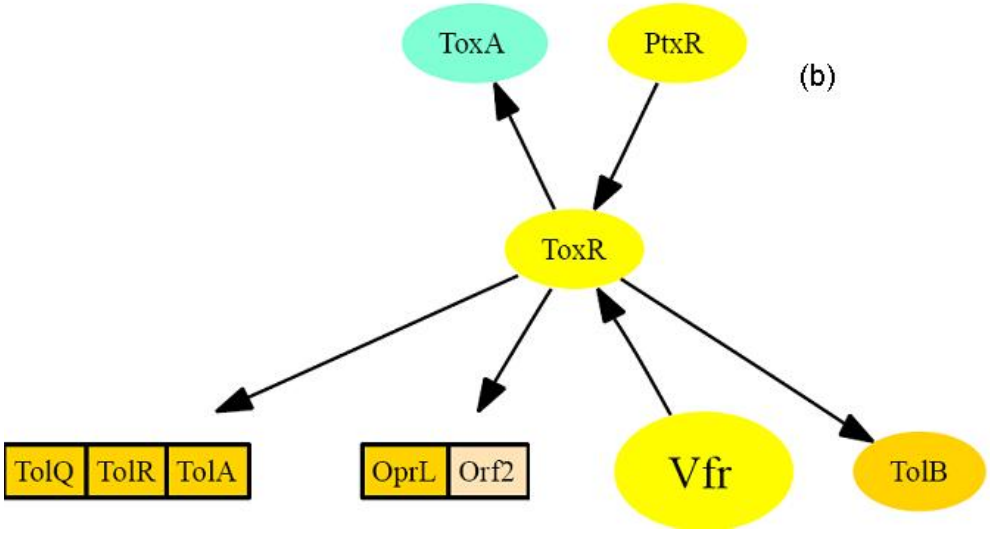


Figure 4-10. Query result for gene "ToxR" (a) detailed page for query gene "ToxR" (b) visualized result for query gene "ToxR".

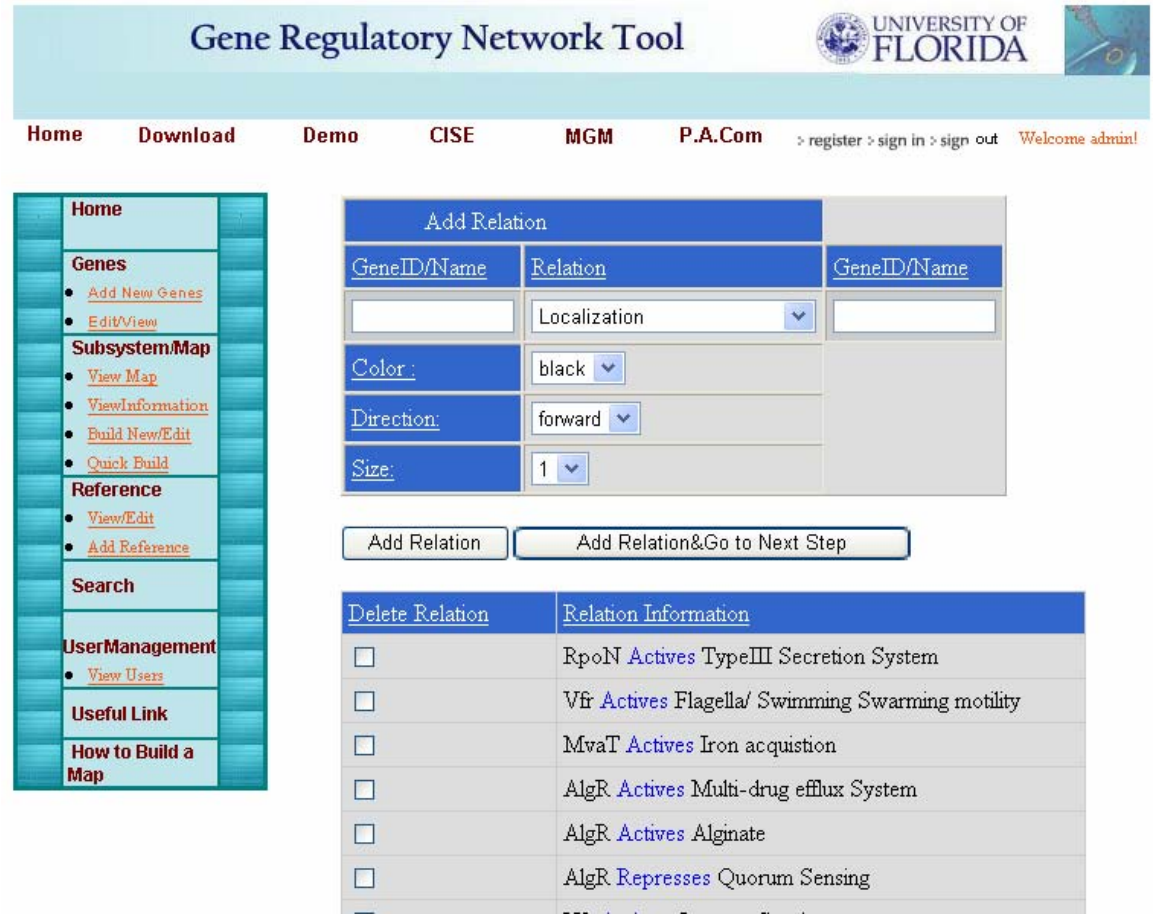


Figure 4-11. Part of interface for online build/edit tool for regulatory networks.

CHAPTER 5
ANALYSIS TOOLS SUBSYSTEM

In this chapter, we introduce the design and implementation of the Analysis Tools Subsystem in GRNS. The Analysis Tools Subsystem includes the frequent pattern mining tool and the gene relation prediction tool.

Frequent Pattern Mining Tool

We first give the problem definition of frequent graph mining. Then we present several data mining algorithms: RNGV (Regulatory Networks Generation Variation) is for regulatory networks generation and variation, DFS (Data Mining Frequent Subgraphs) is for data mining frequent subgraphs, and IGM is for the inexact graph match algorithm. In the end, we provide several frequent patterns from GRNS and KEGG Database.

Problem Definition

We model regulatory networks by directed graphs, which represent the entry elements in vertices and the relations between entry elements as directed edges. We define our goal in mining regulatory networks as discovering a set of frequent subgraphs in a set of regulatory networks. Given a collection of regulatory networks R_1, R_2, …R_n, a support rate Θ and an inexact threshold ϵ, the data mining of frequent subgraphs in a regulatory networks database is to find all closed connected subgraphs that match exactly or inexactly to at least Θ*n of the input of regulatory networks, with the inexact match threshold ϵ.

A closed (or maximal) subgraph is defined as a subgraph whose super graph does not have the same frequency in the database. The inexact match threshold ϵ is defined as: given a subgraph S, a regulatory network R, if we say S is an inexact match to R under a threshold ϵ, then the minimum cost of a sequence of graph edit operations that must be performed on R to produce S will be no larger than Size(S)* ϵ. Size(S) is the number of the vertex in S plus the

number of the edges in S. The graph edit operations include the vertex and the edge deletion, insertion or re-labeling. In our model, the costs of edit operations are defined as follows:

- Insertion cost: equal to 1 for each insertion of vertex or edge.
- Deletion cost: equal to 0 for each deletion of vertex or edge.
- Re-labeling cost: equal to 1 for each re-labeling of vertex or edge.

The notations we use are shown in Table 5-1.

Frequent subgraphs mining in a large regulatory network database is a particularly challenging problem. However, using the unique vertices model, we may simplify the graph mining problem and reduce the exact match problem to the frequent item-sets mining problem. The frequent item-sets mining problem has been well-studied in the data mining field. There are several existing algorithms dealing with frequent item-sets mining. Among them, Apriori is one of the most efficient. Our approach follows the basic structure of the Apriori algorithm, and makes our additional contributions. We make the revisions for several considerations. First, we use the item set instead of the adjacency matrix model representation to save space. Second, the Apriori Algorithm treats the vertex as an item but we treat the directed edge as an item. Besides, the Apriori Algorithm extends one vertex in each round but our approach extends one edge in each round. Extending edges can achieve better efficiency.

Regulatory Network Generation and Variation (RNGV) and Data Mining Frequent Subgraphs (DFS)

RNGV (Regulatory Network Generation and Variation) algorithm reads all input regulatory networks, builds a global index table for all vertices and edges based on the vertex's name, then finds all frequent edges, and calls DFS to extend each edge to a larger size frequent subgraph. RNGV is shown as follows.

Algorithm Name: RNGV

Input: D, n, Θ, ϵ

Output: frequent subgraphs sets F

Read in every Regulatory Network R_i, build a global vertex table and global edge table, store every R_i using global vertex index and global edge index.

For each edge e_i in global edge table

{ if it appears at least n*Θ, let

S_1={ e_i };

Let C_1=Neighbor(e_i);

Call DFS (D, S_1 , C_1 , Θ, ϵ, F);

}

Output frequent subgraphs sets F

DFS is shown as follows. The basic idea is that we treat directed edges as items of traditional frequent item-sets. DFS uses the depth-first algorithm and increases the size of the subgraph by extending an edge each time.

Algorithm Name: DFS

Input: D, S_k , C_k , Θ, ϵ,F

Output: F

for all edges $e_i \in C_k$ do

{

Add e_i to V;

Extend S_k to S_{k+1} by adding edge { e_i };

For every R_j in D, Call IGM (S_{k+1}, R_j, ϵ , f)

if (IGM (S_{k+1}, R_j, ϵ , f) return true in at least Θ*n RegulatorNetworks)

Extend successful;

C_{k+1} =($C_k \cup N(e_i)$) / V;

DFS (D, S_{k+1} , C_{k+1} , Θ, ϵ);

}

if (not extend successful)

{

if S_k is closed then F =F $\cup S_k$;

return F;

}

An example of RNGV and DFS is shown in Figure 5-1. We have a collection of four graphs. Let Θ =0.5, ϵ = 0.25. In RNGV, first, we read in four graphs and build a global index for vertices {a,b,c,d,e,h,m} and directed edges {ab,ac,bc,bd,bh,bm,de}. Then, we find all frequent edges {ab,ac,bc,bd}; call DFS for each edge. In DFS, we use the depth-first algorithm and extend one edge in each time. For edge ab, we can extend successfully to {ab,ac,bc,db},{ab,ac,bc,bh},{ab,ac,bc,bm}. For any other edge, we cannot find other closed subgraphs. In the end, we get the closed subgraph sets {{ab,ac,bc,db},{ab,ac,bc,bh},{ab,ac,bc,bm}}.

Inexact Graph Matching Algorithm (IGM)

By reducing the graph match problem to the data mining frequent item sets problem, we get a simplified method mining exact frequent subgraph. We can efficiently mine exact match frequent subgraphs. However, the utilizing of unique vertex labels is not sufficient for the inexact match. In the inexact match, we allow a sequence of graph edit operations to be applied on regulatory networks. Since the inexact graph isomorphism is an NP-complete problem, heuristic methods are required to reduce the computation time. Our goal here is to efficiently find whether two graphs are an inexact match. We can reduce the computation time by ignoring some unimportant matching detail and delete some non-efficient matching sub-solution. In GRNS, two heuristic methods are proposed to reduce the computation time when computing the inexact match. The first one is the fast checking mechanism to effectively reduce the running time; the second one is the bounded search mechanism to limit the size of the search space.

The idea of the fast checking mechanism is as follows: before the next round of edit operation, we check whether we can finish the computation even if we do not know how these two graphs are isomorphic and further what the ultimate edit operations sequence is like. Before we explain how fast the checking works, let us see the definition of insertion-distance.

The insertion-distance of a subgraph S (Vs, Es) and a Regular Networks R (V, E) (shown as I_dist(S,R)) is defined as follow:

I_dist(S, R)=size(S)–(|Matched(V)|+|Matched(E)|), where Matched (V)=Vs∩V, Matched(E)=Es∩E.

The meaning of Insertion-Distance between S and R corresponds to a sequence of edit operations, each of which is to delete every vertex and edge in R which does not appear in S, and insert the vertex and the edge to R which appear in S but not appear in R. Although the sequence of edit operations corresponding to the Insertion-Distance may not be the optimal edit operations

sequence, we guarantee that a regulator network R (V, E) can produce subgraph S (Vs, Es) under this edit operations sequence. Therefore, we define fast checking in the following way:

A subgraph S is the inexact match to a regulator network R if after a sequence of edit operations $\{o_1, o_2, o_3\ldots\}$ with cost m, we get R' (V', E'). Given a thresholds ϵ, we have: m+ I_dist (S, R') <= ϵ *size (S).

Fast checking can reduce the computation time. The idea behind the fast checking mechanism is, in computation for answering whether S is an inexact match to R, it is not necessary to find an ultimate edit operations sequence. The computation complexity of I_dist (S, R') is | Es |*log |E'| +| Vs | *log |V'| if we use the AVL tree store the graph.

Another mechanism we use to reduce computation time is the bounded search mechanism. We use it to limit the size of the search space. We only keep some good edit operations sequences as the candidates of the next round search, and prune other bad edit operations sequences. Given a bounded search factor f (usually between 0 and 1), the bounded search is described in the following manner:

Given a subgraph S (Vs, Es) and a regulator network R (V, E), after a sequence of edit operations $\{ o_1, o_2, o_3\ldots\}$ in R, we get R' (V', E') and an edit cost m. We define that R' (V', E') and $\{ o_1, o_2, o_3\ldots\}$ can be kept to next round if m + I_dist(S, R') <= f*size(S).

Based on the two heuristic methods introduced previously, we develop the IGM algorithm as follows:

Algorithm Name: IGM

Input: S, R, ϵ , f

Output: true if inexact matched, false if not

```
MAXCOST:= ϵ*size(S);
 if( I_dist(S,M) <= MAXCOST) return true;
 N:= Neighbor(S,R);
 if  N=Φ  return false;
 for every vertex v_i in S
{
  Cost++;  //means the cost for re-labeling =1
    for every vertex n_j in R
     {
   if(vertex in R and vertex vi is equal )
          Cost--;
    if it is the first step mapping
      {
   Mapping the R to R' by relabeling n_j to v_i;
         if(I_dist(S, R') + Cost<= MAXCOST)
                return true;
       if (I_dist(S, R') + Cost<=
                f*size(S))
    add Mq:= ( nj, vi ) to the Queue;  }
      else //it is not the first time mapping
 {
    while(Queue not Empty)
{    read the map from queue in Mq;
```

```
                    if  Mq is a Break Symbol, break;
                    Mq:=Mq∪ ( nj,vi) ;
    Mapping the R to R' by relabeling every
        pair in Mq;
if( I_dist(S,m) + Cost<= MAXCOST) return
        true;
if (I_dist(S,m) + Cost<= f*size(S))
            add  Mq to the Queue;
  } //end of while
 } //end of for every vertex nj in R
   Put a Break Symbol in Queue, so we can
      recognize     when current level finish;

}
```

Here is an example of how IGM works in Figure 5-2. Given Subgraph1 and Graph1, if ϵ = 0.25, IGM returns true in 2nd line. Since I_dist (Graph1, subgraph1) = 2 and ϵ *Size (Subgraph1) = 2. IGM does not need any more computation. If ϵ = 0.2, f = 0.33, we need to do more work. The process is shown in Figure 5-2 (b). The (c,a) means relabeling vertex c in Graph1 to vertex a. (c,a) 6 means if relabeling c in Graph1 to a, I_dist(Subgraph1,Graph1)+cost = 6. In each round, only the ones with a rectangle, which fit the condition of the bounded search, are put in queue and computed in the next round. We discover, using bounded search factor f, we can prune many bad edit operations and only keep some good ones. For example, only (a,a) is

saved in the first round. In the end, we find re-labeling vertex h to d can let I_dist (Subgraph1, Graph1) + cost <= ϵ*Size (Subgraph1). IGM ends here and returns true.

Result

One example of frequent pattern in GRNS

We show the frequent subgraphs in Regulatory Networks in GRNS in Figure 5-3. Right now, no obvious repeated pattern appears in GRNS. Most of the frequent elements are the results we include in other subsystems and their relations between global regulators when we display one subsystem.

One example of frequent pattern in KEGG

The frequent pattern mining tool is a standalone tool. We can use this tool to mine frequent pattern not only in GRNS but also use in other systems. The following example is a frequent pattern we discover in KEGG (Kyoto Encyclopedia of Genes and Genomes) database. KEGG is a database on molecular interaction networks in biological processes. By March of 2005, KEGG includes carbohydrate metabolism, energy metabolism, lipid metabolism, amino acid metabolism and so on, an overall of 15 types. The total number of pathways is 20,560 from 250 organisms.

The results of frequent subgraphs discovered in the reference pathways of amino acid metabolism pathways are shown in Figure 5-4. The vertices of the figures are labeled by the enzymes' name corresponding to the KEGG ID (for detailed information access: http://www.genome.jp/dbget-bin/www_bfind?enzyme).

We find out that, extending graph mining to inexact matching, we can get potentially more interesting results.

Comparison of IGM with SUBDUE

The run time for data mining the amino acid metabolism pathways under different inexact match thresholds is shown in Table 5-2. Our algorithm has satisfactory performance. For this

large dataset, it only takes about one second. Besides, when extending the exact matching to the inexact matching, our computation time does not increase much. The computation time for the inexact matching is comparable to the computation time to Koyutürk et al. (Koyutürk *et al.*, 2004) for the extract matching time. Moreover, since we use sparse graph representation to save the space, our algorithm also shows better space saving compared with other algorithms, for example, the SUBDUE System. We show the run time of IGM and SUBDUE system for the inexact matching in Table 5-3. We find out that, on one hand, IGM is much faster than SUBDUE; on the other hand, when ϵ increases, SUBDUE cannot give us the results due to the run-out memory. However, IGM can give the results, when ϵ increases. IGM is an efficient algorithm and has better performance both in run time and space.

Gene Relation Prediction Tool

We first give the problem definition of gene relation prediction. Then we present the gene prediction algorithm. In the end, we provide an example of the gene relation prediction.

Problem Definition

We define our goal in gene relation predicting algorithm as discovering a complete set of potential gene relations based on a collection of gene regulatory networks and several rules. We model regulatory networks by directed graphs, which represent the entry elements in vertices, the relations between entry elements as directed edges and the type of regulation as labels for edges. We define the gene relation prediction problem as follows:

Suppose we have a collection of gene regulatory networks, represented by directed graphs $\{(V_1, E_1), (V_2, E_2), \ldots(V_n, E_n)\}$. First, we build a new graph G' $=((\bigcup_{i=1}^{i=n} V_i), (\bigcup_{i=1}^{i=n} E_i))$. Then given any vertex *u* in G', the gene regulation relation prediction algorithm tells whether there is a path *p* from *u* to any vertex *w* in the graph G' if *w* is not directly connected with *u*. If *p* exists, then we

store this path. Moreover, based on the label information in every edge in *p*, we label the *p*. The gene regulation relation prediction algorithm is similar to the transitive closure algorithm for a directed graph with the two supplements: first, the gene regulatory relation prediction algorithm stores the path information if the path between the given pair of vertices exists; second, if the path between the given pair of vertices exists, the gene regulation relation prediction algorithm needs to decide the label information for the path.

Gene Relation Prediction Algorithm

Algorithm Name: GRPA

Input: G' =(($\bigcup_{i=1}^{i=n} V_i$), ($\bigcup_{i=1}^{i=n} E_i$)),vertex *u*

Output: A collection of path *p*, which *p* is a path from *u* to any vertex *w* in the graph G' in the case *w* is not directly connected with *u*.

Neighbor_*u* = all vertices directly connected to *u*

newAdd = copy of Neighbor_*u* ;

While newAdd size > 0

{

Edge= copy of newAdd;

newAdd = empty;

For every vertex *i* in Edge

{

Neighbor_*i* = all vertices directly connected to *i*

For every vertex *j* in Neighbor_*i*

{

If vertex *j* is not in Neighbor_*u*

{ Add *j* in newAdd;

Add *j* in Neighbor_*u*;

Add path *p*= {*u*->-> *i*->*j* } to Path_*u* ;

}

}

}

}

For every path *p* in Path_*u*, predict the label for *p*.

Output Path_*u* ;

Result

We show the gene relation prediction algorithm result for the gene "LasR" in Figure 5-5. We find the predicting result includes the discovered potential relations with LasR and the evidence information for every predicting relation.

Table 5-1. Notations in RNGV, DFS and IGM.

Symbol	Description
D	The dataset of a collection of metabolic pathways
n	The numbers of regulator Network in D
Sk	A subgraph with k edges
R	A regulator network
Ck	A set of candidates with k edges
F	A set of frequent subgraphs
V	A set of already visited edges
N(ei)	The neighbor of ei . The set of edge share at least one vertex with ei
ə	Threshold of inexact match
Θ	Support rate threshold
f	bounded search factor

Table 5-2. Run time of IGM on metabolism pathways datasets.

Metabolism Pathway	Support rate threshold Θ	Threshold of inexact match ə	Running Time(sec.)
Amino Acid	15%	0	0.8
Metabolism	15%	0.1	1.0

Table 5-3. Run time comparisons with subsude system.

Metabolism Pathway	Size of Metabolism Pathway	Support rate threshold Θ	Threshold of inexact match ə	Running Time(sec.) In IGM	Running Time(sec.) In Subdue
Amino Acid	179	0.1	0.1	1.0	1000
Metabolism			0.15	1.5	-----

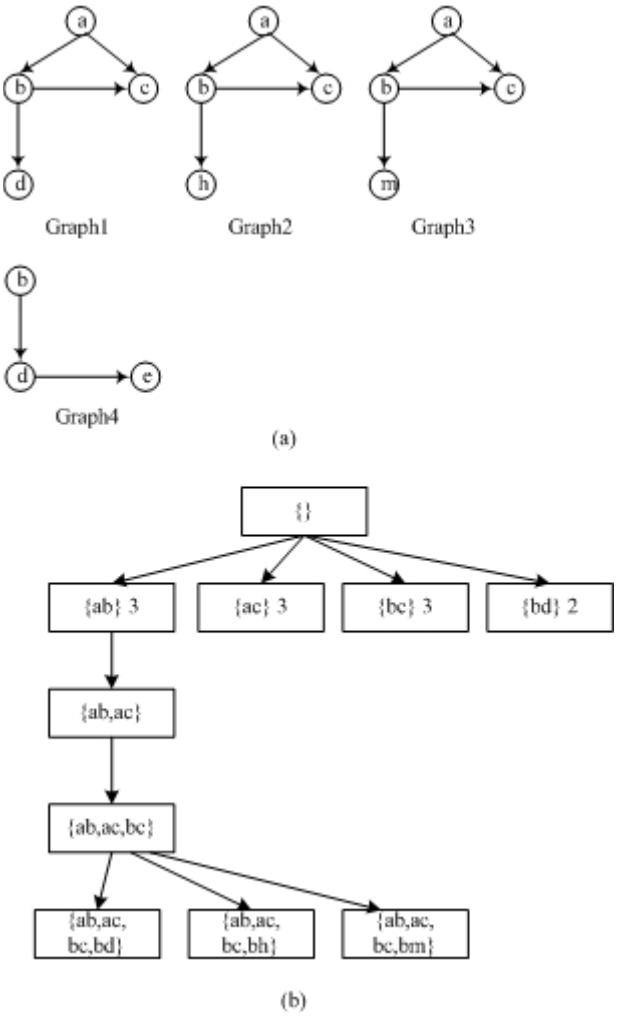


Figure 5-1. An example of RNGV and DFS. (a) a collection of metabolic pathways. (b) the process of RNGV and DFS.

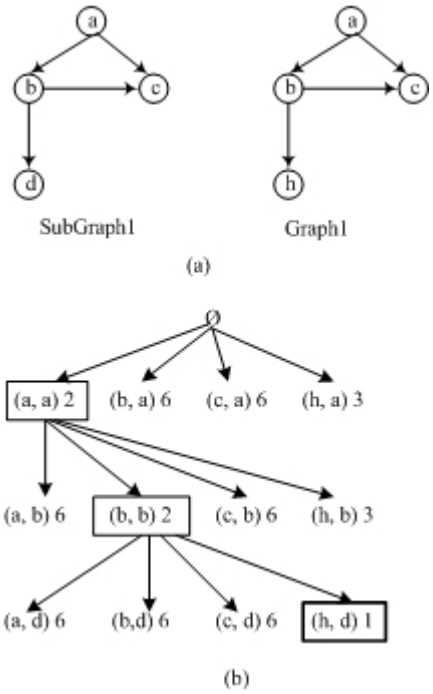


Figure 5-2. An IGM example. (a) subgraph1 and graph1 (b) inexact match process.

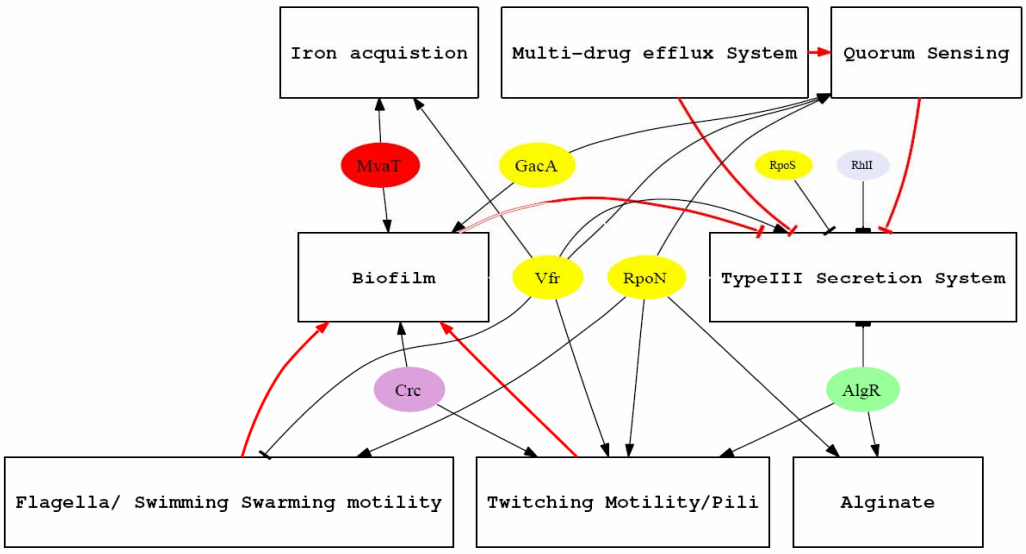


Figure 5-3. Frequent subgraphs in regulatory networks in GRNS when Θ=25% and ϵ =0.

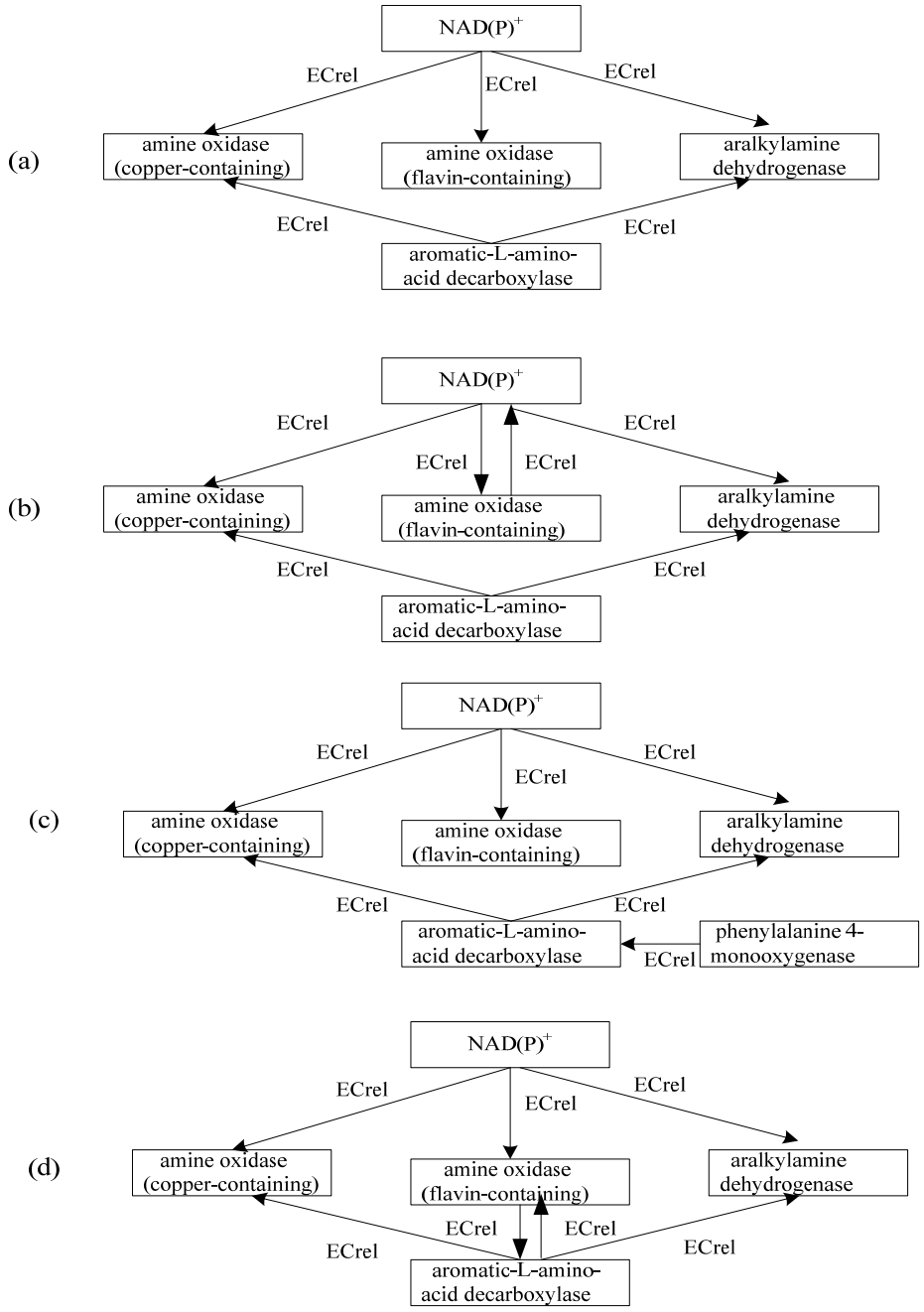
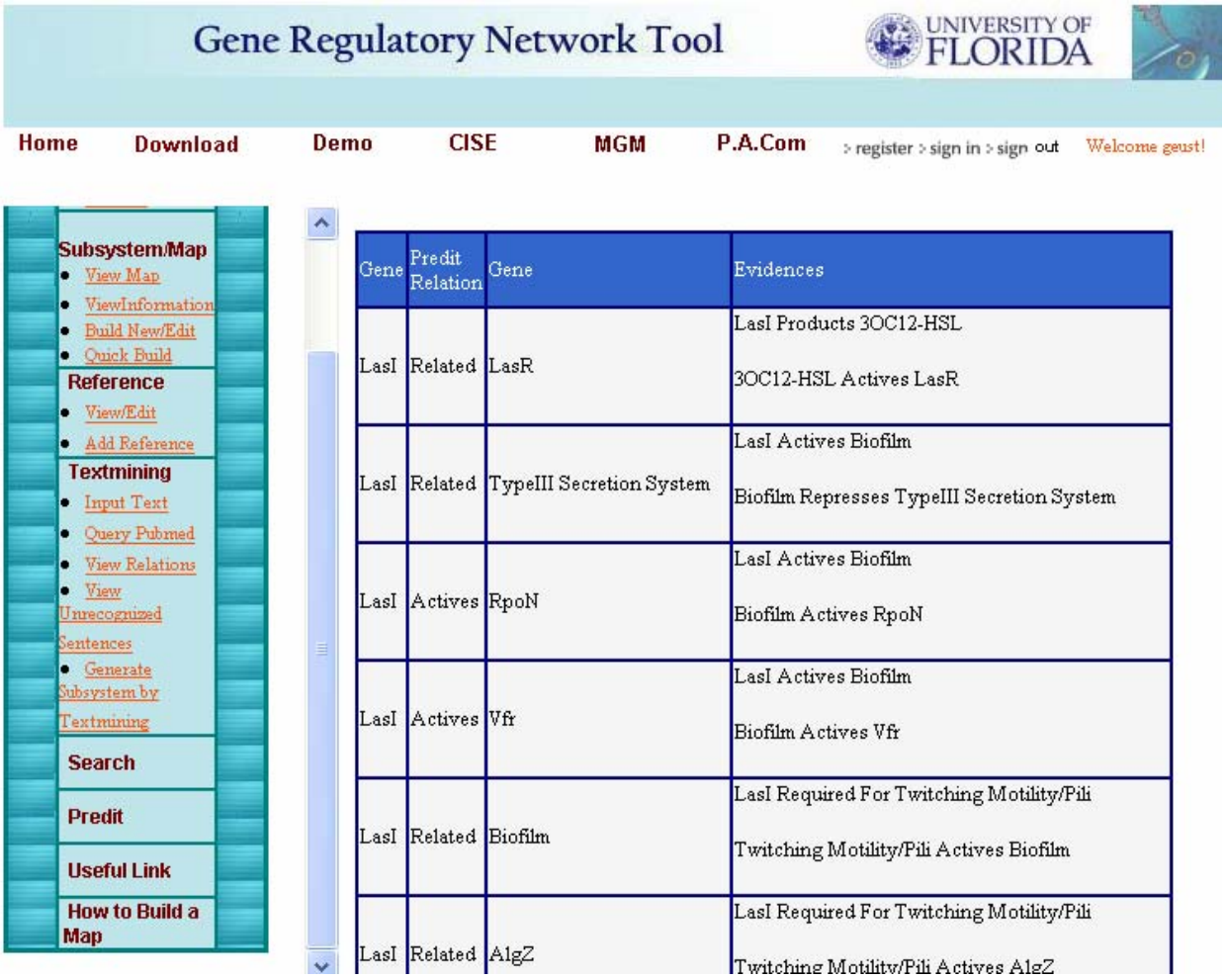


Figure 5-4. Frequent subgraphs in amino acid metabolism: (a) exact matched subgraph; (b), (c), (d), (e) inexact matched subgraph when Θ=15% and ϵ =0.1.



Gene	Predit Relation	Gene	Evidences
LasI	Related	LasR	LasI Products 3OC12-HSL 3OC12-HSL Actives LasR
LasI	Related	TypeIII Secretion System	LasI Actives Biofilm Biofilm Represses TypeIII Secretion System
LasI	Actives	RpoN	LasI Actives Biofilm Biofilm Actives RpoN
LasI	Actives	Vfr	LasI Actives Biofilm Biofilm Actives Vfr
LasI	Related	Biofilm	LasI Required For Twitching Motility/Pili Twitching Motility/Pili Actives Biofilm
LasI	Related	AlgZ	LasI Required For Twitching Motility/Pili Twitching Motility/Pili Actives AlgZ

Figure 5-5. Part of gene regulation prediction results for gene "LasI".

CHAPTER 6
CONCLUSION AND FUTURE WORKS

Conclusion

In this dissertation, we present the framework of the Gene Regulatory Networks System: GRNS. The goals of GRNS are automatically mining biomedical literature and constructing gene regulatory networks based on extracted biomedical information. GRNS first utilizes an automated text mining method to extract useful information from biomedical literature. Then it automatically constructs and visualizes the regulatory networks based on the extracted information and existed domain-specific knowledge. To provide researchers with a clear and global view about the regulatory networks, GRNS uses an interactive visualization method, integrating biomedical evidential information into the regulatory networks. Once a user clicks on an entity or relation of interest, the query interface returns a detailed information page about the clicked entity or relation. It provides researchers with a web-based user-friendly tool to build and edit regulatory networks. GRNS provides a standalone analysis tool to find the frequent patterns in regulatory networks, which can help researchers to find biomedical meaningful patterns. GRNS also provides the gene relation prediction tool to predict potential gene regulatory relation information. In summary, GRNS is a useful tool to automatically extract gene regulatory information from biomedical literature, construct gene regulatory network based on the information extracted, and provide the online management and analysis of gene regulatory networks.

Future Work

Analysis Tool for Error Detecting in Regulatory Networks

In the future, the error detection tool may be developed to find potential errors in the regulatory networks. It will utilize a rule-based method and help to improve the accuracy of the

regulatory networks. One example of a possible rule can be: if one regulatory network includes a relationship “A Products B”, then B must be a product type. If B is any other type, then either the relationship “A Products B” is wrong or the type of B is wrong.

Regulatory Networks Predication Model

GRNS provides the gene relation predication model for potential gene relation information. How about the predication model for an unknown genome data? The text mining method can efficiently construct regulatory networks on well-studied genome data, i.e. if there is enough research literature about this genome field. But if there is a newly discovered genome, how we can quickly predict the functional roles of each gene in this genome and build up the predicted regulatory networks without much literature knowledge?

Our basic idea is in the following: there may be a well-studied genome B similar to the newly discovered genome A. Regulatory networks about B are available. If we can find out a mapping rule about genes in genome A and genes in genome B, then we can get regulatory networks about genome A by replacing all genes in regulatory networks of genome B to mapping genes in genome A.

One possible mapping rule is using BLAST (Basic Local Alignment Search Tool) to find the most similar genes between genome A and genome B. The Basic Local Alignment Search Tool (BLAST) can find regions of local similarity between sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

Although the basic idea seems simple, there are several uncertain factors to the prediction model. First, how does one decide if genome A and genome B are similar enough? Apparently, if we use non-similar genomes in the predicting model, no interesting results will be expected.

Second, how does onbe decide the threshold for the gene similarity? BLAST will calculate the statistical significance of matched sequences. We will need to decide a threshold to assign the result into binary categories. Third, is there any other information that can help us prepare the prediction model?

Regulatory networks predication model is an interesting topic. We may expand our system to make some attempts to build a predication model for regulatory network construction.

LIST OF REFERENCES

Abney, S. (1996) Statistical methods and linguistics. *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, The MIT Press, 1-26.

Aendekerk,S., Ghysels,B., Cornelis,P., Baysse,C. (2002) Characterization of a new efflux pump, MexGHI-OpmD, from *Pseudomonas aeruginosa* that confers resistance to vanadium. *Microbiology*, **148**, 2371-2381.

Afantenos,S., Karkaletsis,V., Stamatopoulos,P. (2005) Summarization from medical documents: a survey. *Artif Intell Med*, **33(2)**, 157-177.

Agrawal,R., Srikant,R. (1994) Fast algorithm for mining association rules. *Proc. of Twentieth Very Large Data Base Conference: VLDB'94*, 487-499.

Agrawal,R., Srikant,R. (1995) Mining sequential patterns. *Proc. Eleventh International Conference on Data Engineering:ICDE'95*, 3-14.

Arora,S.K., Ritchings,B.W., Almina,EC., Lory,S., Ramphal,R. (1997) A transcriptional activator, FleQ, regulates mucin adhesion and flagellar gene expression in *Pseudomonas aeruginosa* in a cascade manner. *Journal of Bacteriology*, **179(17)**, 5574-5581.

Bengio, Y. (1993) A connectionist approach to speech recognition. *International Journal on Pattern Recognition and Artificial Intelligence,* **1**, 425-428.

Blaschke,C., Andrade,MA., Ouzounis,C., Valencia,A. (1999) Automatic extraction of biological information from scientific text: protein-protein interactions. *Proc of Conf Intell Syst Mol Biol*. **7**, 60-67.

Boeckmann,B. Bairoch,A., Apweiler,R., Blatter,M.C., Estreicher,A., Gasteiger,E., Martin,M.J., Michoud,K., O'Donovan,C., Phan,I. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL. *Nucleic Acids Res.,* **31**, 365-370.

Brill,E. (1995) Transformation-based error-driven learning and natural language pro-pessing: a case study in part-of-speech tagging. *Computational Linguistics*, **21(4)**, 543- 566.

Brutlag,D.L., Galper,A.R., Millis,D.H. (1991) Knowledge-based simulation of DNA metabolism: prediction of enzyme action. *Computer Applications in the Biosciences*, **7(1)**, 9-19.

Chakrabarti,S. (2000). Data mining for hypertext: a tutorial survey, *ACM SIGKDD Explorations*. **1(2)**, pp. 1-11.

Chun,H., Tsuruoka,Y., Kim,J., Shiba,R., Nagata,N., Hishiki.T., Tsujie,J. (2005) Extraction of gene-disease relations from medline using domain dictionaries and machine learning. *Proceeding* of *Pac. Symp. Biocomput.,* **11**, 4-15.

Cohen,K., Hunter, L., (2004) Natural language processing and systems biology. *In Dubitzky and Pereira, Artificial intelligence methods and tools for systems biology*. Springer Verlag **5**,147-173.

Cohen,A.M., Hersh, W.R., (2005) A survey of current work in biomedical text mining. *Briefings in Bioinformations.* **6(1)**, 57-71.

Dasgupta,N., Ramphal,R. (2001) Interaction of the antiactivator FleN with the transcriptional activator FleQ regulates flagellar number in *Pseudomonas aeruginosa*. *Journal of Bacteriology*, **183(22)**, 6636-6644.

Dogrusoz,U., Erson,E., Giral,E., Demir,E., Babur,O., Cetintas,A., Colak,R. (2006) PATIKA web: a web interface for analyzing biological pathways through advanced querying and visualization. *J. Bioinformatics.*, **22(3)**, 374-375.

Frawley,W., Piatetsky-Shapiro, G., and Matheus, C. (1992). Knowledge discovery in databases: an overview, *AI Magazine*, 13(3), pp. 57-70.

Fischer,I., Meinl,T. (2004) Graph based molecular data mining--an overview. *International Conference on Systems, Man and Cybernetics: SMC 2004*, 4578-4582.

Friedman,C., Kra,P., Yu,H., Krauthammer,M., Rzhetsky,A. (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformations.,* **17(Suppl 1)**, S74-82.

Friedman,N., Linial,M., Nachman,I., Pe'er,D. (2000) Using bayesian networks to analyze expression data. *In Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, RECOMB, New York, N.Y., ACM Press, 127-135.

Gansner,E.R., Koutsofios,E., North,S.C., Vo,K.P. (1993) A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*. **19(3)**, 214-230.

Gansner,E.R., North,S. (1999) An open graph visualization system and its applications to software engineering. *Software-Practice and Experience*, **30(11)**, 1203-1233.

NIH, (2006) GeneBank, *NIH*, http://www.ncbi.nih.gov/Genbank/, 06/2007 accessed.

Greenberg, E.P. (2000) Bacterial genomics. pump up the versatility. *Nature*, **406(6799)**, 959-964.

Goodman,A.L., Lory,S. (2004) Analysis of regulatory networks in *Pseudomonas aeruginosa* by genomewide transcriptional profiling. *Curr Opin Microbiol*. **7**, 39-44.

Heinrichs,D.E., Poole,K. (1996) PchR, a regulator of ferripyochelin receptor gene (fptA) expression in *Pseudomonas aeruginosa*, functions both as an activator and as a repressor. *J. Bacteriol*. **178(9)**, 2586-2592.

Hirschman,L., Park,J.C., Tsujii,J., Wong,L., Wu,CH. (2002) Accomplishments and challenges in literature data mining for biology. *Bioinformations*. **18(12)**,1553-1561.

Holford,M., Li,N., Nadkarni,P., Zhao,H., VitaPad: visualization tools for the analysis of pathway data. *J. Bioinformatics.*, **21(8)**, 1596-1602.

Hu,ZZ., Narayanaswamy,M., Ravikumar,KE., Vijay-Shanker,K., Wu, CH. (2005) Literature mining and database annotation of protein phosphorylation using a rule-based system. *Bioinformations.,* **21(11)**, 2759-2765.

Inokuchi,A., Washio,T. (2003) Complete mining of frequent patterns from graphs: mining graph data. *J. Machine Learning,* **50**, 321-354.

Jong,H.D. (2003) Modeling and simulation of genetic regulatory systems: a literature review. *J Compute Biol.* **9(1)**, 67-103.

Jyot,J., Dasgupta,N., Ramphal,R. (2002) FleQ, the major flagellar gene regulator in *Pseudomonas aeruginosa*, binds to enhancer sites located either upstream or atypically downstream of the RpoN binding site. *Journal of Bacteriology*, **184(19)**, 5251-5260.

Kanehisa,M. (1996) Toward pathway engineering: a new database of genetic and molecular pathways. *Science & Technology Japan*, **59**, 34-38.

Karp,P., Paley,M., Romero,P. (2002) The pathway tools software. *J. of Bioinformatics.*, **Suppl 18(1)**, S225-S232.

Kauffman,S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, **22**, 437-467.

Koyutürk,M., Grama,A., Szpankowski,W. (2004) An efficient algorithm for detecting frequent subgraph in biological networks. *ISMB/ECCB*, i200-207.

Krishnamurthy,L., Nadeau,J., Özsoyoglu,G., Özsoyoglu,Z., Tasan,M., Xu,W. (2003) Pathways database system: an integrated system for biological pathways. *J. Bioinformatics.*, **19(8)**, 930-937.

Kuramochi,M., Karypis,G. (2004) Finding frequent patterns in a large sparse graph. *Data Mining and Knowledge Discovery*, **11(3)**, 243-271.

Larbig,K., Kiewitz,C., Tummler,B. (2002) Pathogenicity islands and PAI-like structures in *Pseudomonas* species. *Curr Top Microbiol Immunol.,* **264(1)**, 201-211.

Lawrence,B., Cook,J. (1993) Discovery of inexact concepts from structural data. *IEEE Trans. Knowl. Data Eng.* **5(6)**, 992-994.

Ling,X., Jing,J., He,X., Mei,Q., Zhai,C., Schatz,B. (2006) Automatically generating gene summaries from biomedical literature. *Proceeding* of *Pac. Symp. Biocomput.,* **11**, 41-50.

Liu,H., Friedman,C. (2003) Mining terminological knowledge in large biomedical corpora. *Proceeding* of *Pac. Symp. Biocomput.,* **8**, 415-426.

Marcotte,EM., Xenarios,I., Eisenberg,D. (2001) Mining literature for protein-protein interactions. *Bioinformations.,* **17(4)**, 359-63.

Mattick, J.S. (2000) Type IV pili and twitching motility. *Annu Rev Microbiol*, **56**, 289-314.

McDonald,DM., Chen,H., Su,H., Marshall,BB. (2004) Extracting gene pathway relations using a hybrid grammar: the Arizona Relation Parser. *Bioinformations.,* **20(18)**, 3370-3378.

Nenadic,G., Spasic,I., Ananiadou,S. (2003) Terminology-driven mining of biomedical literature. *Bioinformations.* **19(8)**, 938-943.

Pinter,RY., Rokhlenko,O., Yeger-Lotem,E., Ziv-Ukelson,M. (2005) Alignment of metabolic pathways. *Bioinformatics*. **21(16)**, 3401-3408.

Rossolini,G.M., Mantengoli,E.(2005) Treatment and control of severe infections caused by multiresistant *Pseudomonas aeruginosa*. *Clin Microbiol Infect*, **11(Suppl 4)**, 17-32.

Saric,J. Jensen,LJ., Ouzounova,R., Rojas,I., Bork,P. (2005) Extraction of regulatory gene/protein networks from Medline. *Bioinformations,* **22(6)**, 645-650.

Sartipi,K., Kontogiannis,K. (2001) A graph pattern matching approach to software architecture recovery. *International Conference on Software Maintenance: ICSM'01*, 408-419.

Serino,L., Reimmann,C., Visca,P., Beyeler,M., Chiesa,V.D., Haas,D. (1997) Biosynthesis of pyochelin and dihydroaeruginoic acid requires the iron-regulated pchDCBA operon in *Pseudomonas aeruginosa. Journal of Bacteriology*, **179**, 248-257.

Serov,V., Spirov,A., Samsonova,M. (1998) Graphical interface to the genetic network database GeNet. *J. Bioinformatics.*, **14(6)**, 546-547.

Shatkay,H., Feldman,R. (2003) Mining the biomedical literature in the genomic era: an overview. *J. Comput. Biol*., **10(6)**, 821- 55.

Shmulevich,I., Dougherty,E.R., Zhang,W. (2002) From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, **90(11)**, 1778-1792.

Shmulevich,I., Dougherty,E.R, Kim,S., Zhang,W. (2002) Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18(2)**, 261-274.

Sirava,M., Schäfer,T., Eiglsperger,M., Kaufmann,M., Kohlbacher,O., Bornberg-Bauer,E., Lenhof,H. (2002) BioMiner-modeling, analyzing, and visualizing biochemical pathways and networks. *J. Bioinformatics.*, **Suppl 18(2)**, S219-S230.

Stover,C.K., Pham,X.Q., Erwin,A.L., Mizoguchi, S.D., Warrener,P., Hickey,M.J., Brinkman,F.S., Hufnagle,W.O., Kowalik,D.J., Lagrou,M., Garber,R.L., Goltry,L., Tolentino,E., Westbrock-Wadman,S., Yuan,Y., Brody,L.L., Coulter, S.N., Folger,K.R., Kas,A., Larbig,K., Lim,R., Smith,K., Spencer,D., Wong,G.K., Wu,Z., Paulsen,I.T., Reizer,J., Saier,M.H., Hancock,R.E., Lory,S., Olson,M.V. (2000) Complete genome sequence of *Pseudomonas aeruginosa* PA01, an opportunistic pathogen. *Nature*, **406**, 959-964.

Todar, K., (2006) Todar's Online Textbook of Bacteriology, *University of Wisconsin-Madison*, www.textbookofbacteriology.net, 06/2007 accessed.

Woods,D.E., (2004) Comparative genomic analysis of *Pseudomonas aeruginosa* virulence. *Trends Microbiol*, **12(10)**, 437-9.

Wu,W., Song,Y., Jin,S., Chen,S. (2005) An interactive map of regulatory networks of *Pseudomonas aeruginosa* genome. *Proceeding* of *First RECOMBS Satellite Workshop on Systems Biology*, San Diego, CA .

Wu,W., Song,Y., Jin,S., Chen,S. (2006) A *Pseudomonas aeruginosa* gene regulation database with a regulatory network map. *Annual Conference American Microbiology Society*, Orlando, FL.

Yandell,M.D., Majoros,W.H. (2002) Genomics and natural language processing. *Nature Reviews Genetics*, **3(8)**, 601-610.

Yakushiji,A., Tateisi,Y., Miyao,Y., Tsujii,J. (2001) Event extraction from biomedical papers using a full parser. *Proceeding* of *Pac. Symp. Biocomput.,* 408-419.

Yuan, X., Hu,ZZ., Wu,HT., Torii, M., Narayanaswamy, M., Ravikumar, KE., Vijay-Shanker, K., Wu, CH. (2006) An online literature mining tool for protein phosphorylation. *Bioinformations.,* **22(13)**, 1668-1669.

BIOGRAPHICAL SKETCH

Ms. Yongling Song is a Ph.D. student in the Department of Computer and Information Science and Engineering at the University of Florida. She enrolled into the Computer Science Department of Xi'an Jiaotong University (XJTU), China in September 1996. The university honored her by waiving the mandatory National University Entrance Exam of China. In 2000, she received her Bachelor of Engineering from the CS department of XJTU. Because of her outstanding scores, in the year 2000, she was honored again when the mandatory Graduate School Entrance Exam of China was waived, and she enrolled in the graduate program of the CS department in XJTU. In May 2003, she received a master's degree from the Computer Science Department at XJTU. She started her Ph.D. studies in the Computer Science Department, at the University of Tennessee at Knoxville in August 2003. In 2004, she transferred to the Department of Computer and Information Science and Engineering at the University of Florida and continued her Ph.D studies under the supervision of Dr. Su-Shing Chen. Her research interests include bioinformatics and data mining algorithms. She received her Ph.D in August 2007.