PathMeld: A Methodology for the Unification of Metabolic Pathway Databases

Harsha K. Rajasimha

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
Computer Science and Applications

Lenwood S. Heath, Ph.D., Chairman

Naren Ramakrishnan, Ph.D.

Ruth Grene, Ph.D.

December, 2004
Blacksburg, Virginia

Keywords: EcoCyc, integration, KEGG, MetaCyc, PathMeld, PGDB, unification

PathMeld: A Methodology for the Unification of Metabolic Pathway Databases

by

Harsha K. Rajasimha

Committee Chairman: Lenwood S. Heath, Ph.D.

Computer Science and Applications

(ABSTRACT)

A biological pathway database is a database that describes biochemical pathways, reactions, enzymes that catalyze the reactions, and the substrates that participate in these reactions. A pathway genome database (PGDB) integrates pathway information with information about the complete genome of various sequenced organisms. Two of the popular PGDBs available today are the Kyoto Encyclopedia of Genes and Genomes (KEGG) and MetaCyc. The proliferation of biological databases in general raises several questions for the life scientist. Which of these databases is most accurate, most current, or most comprehensive? Do they have a standard format? Do they complement each other? Overall, which database should be used for what purpose? If more than one database is deemed relevant, it is desirable to have a unified database containing information from all the shortlisted databases. There is no standard methodology yet for integrating biological pathway databases and, to the best of our knowledge, no commercial software that can perform such integration tasks. While XML based pathway data exchange standards such as BioPAX and SBML are emerging, these do not address the basic problems such as inconsistent nomenclature and substrate matching between databases in the unification of pathway databases.

Here, we present the PathMeld methodology to unify KEGG and MetaCyc databases starting from their flat files. Individual PGDBs are transformed into a unified schema that we design. With individual PGDBs in the common unified schema, the key to the PathMeld methodology is to find the entity correspondences between the KEGG and MetaCyc substrates. We present a heuristic-driven approach for one-to-one mapping of the substrates between KEGG and MetaCyc. Using the exact name and chemical formula match criteria, 82.6% of the substrates in MetaCyc were matched accurately to corresponding substrates in KEGG. The substrate names in the MetaCyc database contain html tags and non-characters such as <sub>, <sup>, <i>, <l>, &, and $. The MetaCyc

chemical formula are stored in lisp format in the database while KEGG stores them as continuous strings. Hence, we subject MetaCyc chemical formulae to transformation into KEGG format to make them directly comparable. Applying pre-processing to transform MetaCyc substrate names and formulae improved substrate matching by 2%. To investigate how many of the remaining 17.4% substrates are indeed absent from KEGG, we employ a standard UNIX based approximate string matching tool called agrep. The resulting matches are curated into four mutually exlusive groups: 3.83% are correct matches, 3.17% are close matches, and 7.45% are incorrect matches. 3.68% of MetaCyc substrate names are not matched at all. This shows that 11.13% of MetaCyc substrate names are absent in KEGG. We note some of the implementation issues we solved. First, parsing only one flat file to populate one database table is not sufficient. Second, intermediate database tables are needed. Third, transformation of substrate names and chemical formula from one of the component databases is required for comparison. Fourth, a biochemist's intervention is needed in evaluating the approximate substrate matches from agrep.

In conclusion, the PathMeld methodology successfully unifies KEGG and MetaCyc flat file databases into a unified PostgreSQL database. Matching substrates between databases is the key issue in the unification process. About 83% of the substrate correspondences can be computationally achieved, while the remaining 17% substrates require approximate matching and manual curation by a biochemist. We presented several different techniques for substrate matching and showed that about 10% of the MetaCyc substrates do not match and hence are absent from KEGG.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The life science community has generated tremendous quantities of data and created numerous biological databases in the last decade. A multitude of these biological databases provide useful information about nucleic acids, proteins, transcription factor binding sites, and motifs.

Based on the categorization of biological databases by Wittig, *et al.* [66], we categorize biological databases based on content as follows:

1. Genome databases such as the genome database (GDB), the TIGR microbial genome database, and GenBank;

2. Protein sequence databases such as SWISS-PROT, and the expert protein analysis system (ExPASy) [17];

3. Enzyme databases such as the ENZYME, the enzyme structure database [1], and the protein data bank (PDB);

4. Interactions databases such as DIP: the database of interacting proteins developed by Xenarios, *et al.* [22], BIND, the biomolecular interaction network database developed by Bader, *et al.* [18], and GRID, a database of genetic and physical interactions.

5. Literature databases such as Pubmed and Medline;

6. Pathway databases such as KEGG and BioCyc [25, 32]; and

7. Other specialized databases such as protein profiles and patterns databases (Prosite [16] and Pfam [5]).

A biological pathway database is a database that describes biochemical pathways, reactions, enzymes that catalyze the reactions, and the substrates that participate in these reactions. A pathway genome database (PGDB) integrates pathway information with information about the complete genome of various sequenced organisms. Most of the pathway databases created thus far describe metabolic pathways. More recently, there is interest in signaling pathway databases that describe the various events that occur when a cell is exposed to different stresses such as UV radiation, heat shock, and drought. Signaling pathway databases differ from metabolic pathway databases in content, structure, and purpose. Karp, *et al.* [33] note that it is only recently that signaling pathways and genetic regulatory pathway databases are emerging. The focus of this research is on metabolic pathway databases. A PGDB serves as a reference for pathway information that can be queried by scientists who want to search out specific facts or patterns. As Karp [26] points out, a PGDB eases the cognitive overload by allowing the life scientist to think in terms of hundreds of pathways rather than in terms of thousands of gene products. One of the primary advantages of studying biological pathways is that it provides descriptions of how the molecular components encoded in a genome interact with each other and with other molecular players to form the biochemical basis of cellular function [46].

Two popular PGDBs are the Kyoto Encyclopedia of Genes and Genomes (KEGG) and EcoCyc. Kanehisa, *et al.* [24] develop KEGG, a bioinformatics resource for understanding cell function and behavior from its genome information. EcoCyc is an organism specific PGDB that describes the metabolic and signal transduction pathways of *Escherichia coli*, its enzymes, its transport proteins, and its mechanisms of transcriptional control of gene expression.

Figure 1.1 shows a PGDB schema as described in the pathway tools user guide [46]. A PGDB schema describes pathways in terms of these five biological entities:

1. Metabolic overview: the union of all described pathways;
2. Pathways: the individual pathways;
3. Reactions: the reactions within these pathways;
4. Compounds: the compounds that participate in these reactions; and

5. Gene products: enzymes that form a subset of gene products and that catalyze these reactions.

The various genes in the nucleus of the cell that constitute the genome are transcribed to produce gene products. A map of various genes and their chromosomal loci form the genomic map. The genes, their products, and the genomic map form the genomic component (enclosed in the smaller dotted rectangle in Figure 1.1). The metabolic overview, pathways, reactions, compounds, and the gene products together constitute the pathway component (enclosed in the larger dotted rectangule in Figure 1.1). Types of pathways that can be incorporated in this scheme include biosynthesis, degradation, energy metabolism, and intermediary metabolism for compounds such as amino acids, carbohydrates, fatty acids, nucleotides, and enzyme cofactors. For our purposes, a genome consists of the following three biological entities:

1. Genomic sequences;

2. Constituent genes; and

3. Gene products that link genes and pathways.

It should be noted that PGDBs typically describe all known or predicted gene products and not just enzymes.

The proliferation of biological databases in general raises several questions for the life scientist. Which of these databases is most accurate, most current, or most comprehensive? Do they have a standard format? Do they complement each other? Overall, which database should be used for what purpose? If more than one database is deemed relevant, it is desirable to have a unified database containing information from all the shortlisted databases. While XML based pathway data exchange standards such as BioPAX and SBML are emerging, these do not address the basic problems such as inconsistent nomenclature and substrate matching between databases in the unification of pathway databases. a model organism database (MOD) is a candidate, such as EcoCyc for *Escherichia coli*. In the case of biological sequence databases, only recently has the need been identified for a central comprehensive repository of information for an organism. OWL, NRDB, and UniProt are some examples discussed in more detail in Chapter 3. These initiatives resulted in more than one such integrated repository, each with its own format and criteria. With biological information being generated in a distributed fashion world wide, it is not feasible to maintain a single database that is absolutely current. Distributed data access is certainly needed. However, it may be desirable

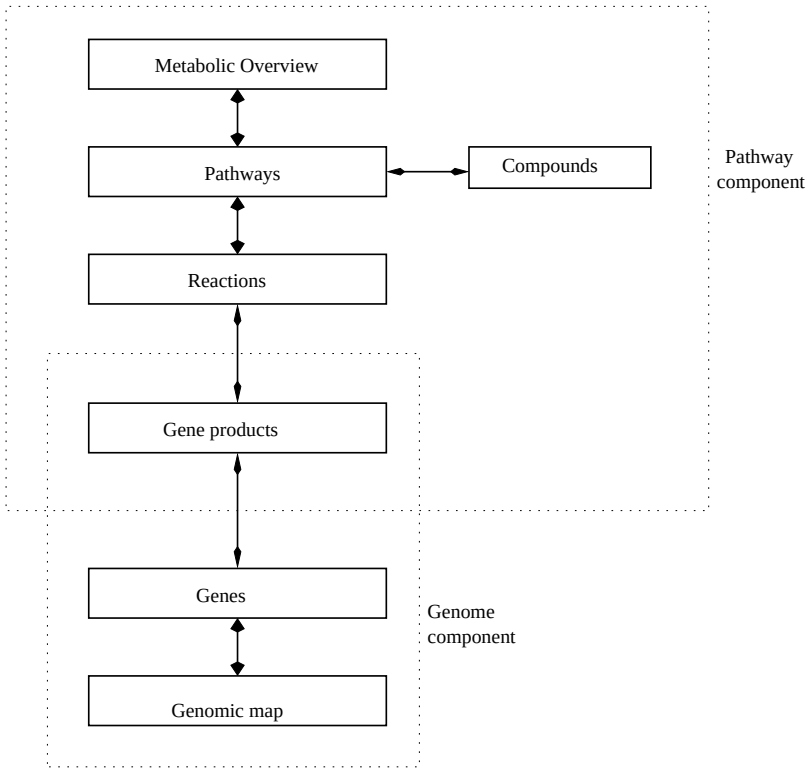

Figure 1.1: A representation of PGDB (taken from the EcoCyc user guide [46]). The genes, their products, and the genomic map form the genomic component (enclosed in the smaller dotted rectangle). The metabolic overview, pathways, reactions, compounds, and the gene products together constitute the pathway component (enclosed in the larger dotted rectangle).

to have comprehensive knowledge of a particular type (for example, pathways of various species) to support biological inferences. Integrating different types of biological databases and integrate several biological databases containing related information require different approaches. Both are enormous tasks in terms of time, effort and computing resources. The problem of interest here can be more generally stated (in the database world) as the problem of integrating heterogeneous databases containing related information [37, 57] . This task is by no means easy, for several reasons, including the following:

1. In data from multiple sources, terminology varies with source.
2. The meaning of an attribute in a database may be ambiguous.
3. Synonyms, aliases, and homonyms are common. Li and Clifton [36] point out that even using the same terms to describe concepts does not assure consistency of factual information in databases.

Saqi and Sternberg [53] develop a structural census of metabolic networks for E.*coli* starting from the metabolic pathway information in KEGG. If a consensus pathway is arrived at, collectively derived from all experimental data from various reliable sources, a globally acceptable structural and functional consensus of all sequenced organisms is obtained. Idekar, *et al.* [23] define systems biology as the study of biological systems under perturbation (biologically, genetically, or chemically); monitoring the gene, protein, and informational pathway responses; integrating these data; and ultimately, formulating mathematical models that describe the structure of the system and its response to individual perturbations. Systems biology describes the work of numerous researchers and is one area where innovative computational techniques are needed. For systems biology progress, easy access to full biological knowledge is required. Typically, this knowledge is either widely distributed or unavailable.

Current PGDBs have limitations. Ahn, *et al.* [2] point out that pathway databases cannot express cell-specific information in eukaryotes. For example, most databases do not provide information about what tissue a particular cell belongs to (e.g., liver or skin) or even the cell phenotype (e.g., cancerous, normal). Ahn, *et al.* [2] suggest that this may be due to the complexity of the 'gene to protein network' in higher eukaryotes. They also make an effort to overcome this limitation by utilizing the notions of semantic object and network. But this is just the start and much still remains a challenge in this direction. The objectives of pathway bioinformatics include construction of

integrated PGDBs, genome scale metabolic pathway reconstruction methods *in silico*, and utilizing such information to build simulation models for further analysis. Current knowledge is often insufficient to build complex biological simulation models. Biological simulations have limitations due to the fact that some links in pathways, or even complete pathways, are still unknown and kinetic models require many parameters that have not been measured in the lab, and cannot be estimated accurately from basic principles. For example, models of biochemical pathways require quantitative information such as physiological concentrations of metabolites and enzymes in specific species. Such data can be obtained only by biochemical experimentation that is often time consuming.

A metabolic pathway is composed of several biological entities such as genes, substrates, enzymes, and reactions. The relationships among these entities are not obvious and have been represented differently in KEGG and MetaCyc. We demonstrate these differences in detail while analyzing the individual database designs. Most biological databases store information related to some of these entities. For example, a genome database contains information that can be viewed as attributes of genes (chromosome positions, sequence annotated function, and species). Similarly, protein and enzyme databases are repositories of information related to the properties of proteins and enzymes such as their subcellular location, and the reactions that they catalyze. Hence, a clear understanding of how these basic building blocks are related will help the automation of distributed database access. We identify such key relationships and show why simplifications are sometimes necessary, depending on the purpose of the database. With the current inconsistencies in naming, we show that matching corresponding substrates forms the key step in the unification and present a heuristics-driven approach to significantly alleviate the problem.

The goal of this research is to design and implement a methodology to integrate and utilize the biological content of different PGDBs. We present a unification methodology called PathMeld for integrating two particular PGDBs, namely, KEGG and MetaCyc. The PathMeld methodology takes advantage of the universal relationships among the various entities involved in a PGDB, namely, compound, reaction, enzyme, gene, and pathway. The idea of PathMeld is to map each of the KEGG and MetaCyc databases to a generalized PGDB schema that is designed to accommodate the required data and their interrelationships. This involves the design of the generalized PGDB schema, database creation, downloading and analyzing the flat files, parsing the flat files to derive data required for database population in tab delimited format, and populating the database tables

with the data. Now, since both databases are transformed to the same schema, matching each entity by their primary key forms the crucial step in their integration. Querying such a system is much easier than a corresponding data-warehouse (a consolidated view of the data from several databases, optimized for reporting and analysis) design because, each of the member databases follow the same schema, table, and entity names, and all part of a single database. We identify key issues involved in the process of integrating biological pathway databases. First, we use EC numbers to map genes onto metabolic pathway diagrams. KEGG associates EC numbers with the enzymes that catalyze reactions of a particular type. On the other hand, MetaCyc associates EC numbers directly with the reactions. This inconsistency is closely related to the distinction of reaction and enzymatic reaction as separate relations in MetaCyc. We note that, in addition to sub-cellular location information, tissue specific information may sometimes be needed for accurate interpretation of the biological pathway. We also list the various issues we resolved during the implementation of PathMeld. Further, recommendations are required to eliminate any scope for errors and inconsistencies in naming entities such as substrates, genes, and reactions. In particular, we call upon the biological database community to strictly adhere to the current standards in nomenclature such as IUPAC (International Union of Pure and Applied Chemistry) for chemical compounds and IUBMB (International Union of Biochemistry and Molecular Biology) for enzymes. This will aide in the construction and analysis of various biochemical pathways for different species.

The rest of the document is organized as follows. Chapter 2 describes some biological pathway and genome databases. Chapter 3 provides a detailed review of biological database integration initiatives. Chapter 4 details the specific objectives of this research. Chapter 5 presents the PathMeld methodology for the unification of pathway genome databases. The chapter describes in detail the implementation of the PathMeld methodology with KEGG and MetaCyc databases. Chapter 6 presents the results of KEGG and MetaCyc unification. The chapter compares KEGG and MetaCyc, both with respect to their biological content and computational design, and concludes with a case study of the glycolysis pathway. Chapter 7 concludes with a discussion of future work in this area.

Chapter 2

Background Information

This chapter briefly describes some of the popular biological pathway and genome databases and their content. A biological pathway may be classified as a metabolic or a signaling pathway. Metabolic pathway databases such as KEGG, MetaCyc, Brenda, WIT, and ExPASy are central to this research and are described in Section 2.1. Signaling pathway databases are discussed briefly in Section 2.2. Given the enormous amounts of data resulting from genome sequencing, functional annotations, and other experimental methods, the list below is inevitably incomplete. Baxevanis [6] provides a recent listing of molecular biology databases.

2.1 Metabolic Pathway Databases

This section introduces metabolic pathway databases that store information about metabolic pathways, reactions, reactants, and the relationships among genes, enzymes, and reactions. Wixon [67] and Wittig, *et al.* [66] provide detailed reviews of most metabolic pathway databases and their websites.

2.1.1 KEGG

The Kyoto Encyclopedia of Genes and Genomes (KEGG) is a suite of databases and associated software, integrating the current knowledge on metabolic networks (the PATHWAY database), genomic and proteomic information (the GENES/SSDB/KO databases), and information about chemical

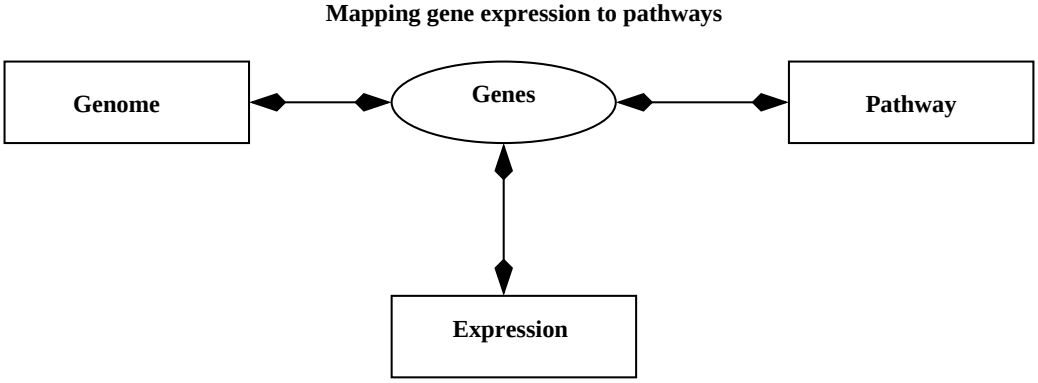

Figure 2.1: The KEGG Database organization.

compounds and reactions (the COMPOUND/REACTION databases).

The KEGG system is organized into tightly connected databases as follows:

1. EXPRESSION database contains microarray gene expression data and information about individual spots.

2. GENES: Gene sequence and information on genes of all completely sequenced organisms and some partially sequenced organisms.

3. LIGAND: Information about over ten thousand chemical compounds, enzyme molecules, and enzymatic and non-enzymatic reactions.

4. PATHWAY: Diagrams of metabolic/regulatory pathways.

The chemical structures of compounds are stored as GIF images and as 2D coordinates stored in an MDL-MOL file (a specific file format for molecular structures from MDL Information Systems Inc.). Either of the two file formats can be used to launch an appropriate drawing application. In the flat file downloadable version, there are files of 7 different types (files with extensions: orth, html, gif, gene, coord, conf, and tab) organized into species-specific directories. The database organization used in KEGG for mapping gene expression data onto pathway diagrams is shown in Figure 2.1.

In addition to the above databases, KEGG provides many links to other databases that are integrated within its database retrieval system called DBGET. The KEGG genes database contains

```
ENTRY       351               CDS       H.sapiens
NAME        APP
DEFINITION  amyloid beta (A4) precursor protein (protease nexin-II, Alzheimer
            disease) [SP:A4_HUMAN]
CLASS              Human Diseases; Neurodegenerative Disorders; Alzheimer's disease
            [PATH:hsa05010]
POSITION    21q21.3
DBLINKS     LocusLink: 351
            GDB: 119692
            OMIM: 104760
            NCBI: 4502167
CODON_USAGE       T               C               A               G
       T   9  12   3  10  10   7   5   1   7  13   0   1   5  13   0   9
       C   7  12   2  22  14   9  10   2  14  11   7  29   3  11   8   5
       A  10  13   1  24   5  27  14   4  12  19  15  26   9   3   7   3
       G  15  10   8  32  15  30  15   3  30  20  44  48   9  12   8   9

AASEQ       770
            MLPGLALLLLAAWTARALEVPTDGNAGLLAEPQIAMFCGRLNMHMNVQNGKWDSDPSGTKTC
            IDTKEGILQYCQEVYPELQITNVVEANQPVTIQNWCKRGRKQCKTHPHFVIPYRCLVGEFVSD
            ALLVPDKCKFLHQERMDVCETHLHWHTVAKETCSEKSTNLHDYGMLLPCGIDKFR AA sequence
            cut here.

NTSEQ       2313
            atgctgcccggtttggcactgctcctgctggccgcctggacggctcgggcgctggaggtacccactgat
            ggtaatgctggcctgctggctgaaccccagattgccatgttctgtggcagactgaacatgcacatgaat
            gtccagaatgggaagtgggattcagatccatcagggaccaaaacctgcattgataccaaggaaggcatc
            ctgcagtattgccaagaagtctaccctgaactg NT sequence cut here.
```

Figure 2.2: A sample entry from the KEGG database.

all publicly available nucleotide sequences and their functional annotations. A sample entry from the GENES database is shown in Figure 2.2.

The KEGG data is in accordance with two international standards IUPAC and IUBMB. The KEGG web service provides access to information such as: metabolic or regulatory pathways, pathways for a specific species, for specific enzymes, compounds, or reactions. However, it does not provide complex query options. For example, the user cannot ask for all enzymes in species X, Y, and Z participating in pathways a, b, or c. KEGG provides several tools for accessing pathways. Kanehisa, *et al.* [54, 24, 14, 43] provide a comprehensive treatment of KEGG tools.

2.1.2 EcoCyc and MetaCyc

Karp, *et al.* [33] describe EcoCyc as an organism-specific pathway and genome database that includes the metabolic and signal-transduction pathways of E.*coli K-12*, its enzymes, its transport proteins, and its mechanisms of transcriptional control of gene expression. EcoCyc is freely available on the web [62]. The MetaCyc database is based on the same database schema as Ecocyc. The MetaCyc database initially contained only the information in the Ecocyc database and subsequently extended with information about more than hundred different species. These databases contain extensive references to literature citations on enzymes and reactions whenever available. The species attribute for each pathway in the database lists all species in which a particular pathway has been cited to be observed in the literature. Hence, absence of a species reference for a pathway does not imply that the pathway does not exist in that species.

The MetaCyc database contains information about all enzymes, reactions, and metabolic pathways of a variety of other organisms with a microbial focus [32]. Since the design of the MetaCyc database and the pathway tools software are completely based on the Ecocyc design, the two resources essentially go together. BioCyc is a collection of several MODs, including EcoCyc, available at the URL [58]. MetaCyc is not merely a collection of related reaction steps from different organisms but is a set of complete pathway information elucidated in specific organisms. Maranas and Burgard [40] note that MetaCyc also provides a wealth of literature citations and in-depth commentary on each enzyme and pathway. All these databases can be accessed using a software environment called pathway tools, which provides querying, editing, and visualization capabilities [32]. The top level hierarchy [28] that forms the basis of the MetaCyc design is shown in Figure 2.3.

A limitation is that one database cannot possibly encompass the complete metabolic picture of every sequenced organism [40]. Following are the intended uses of MetaCyc:

1. As a resource for analysis of microbial genomes at the level of individual genes and an accurate reference for inferring gene function by sequence similarity,

2. MetaCyc describes the subunit structures of many enzymes, and therefore can be used as training or validation datasets for algorithms that depict protein-protein interactions,

3. MetaCyc can serve as a test set for algorithms that infer genetic networks from gene expression data,

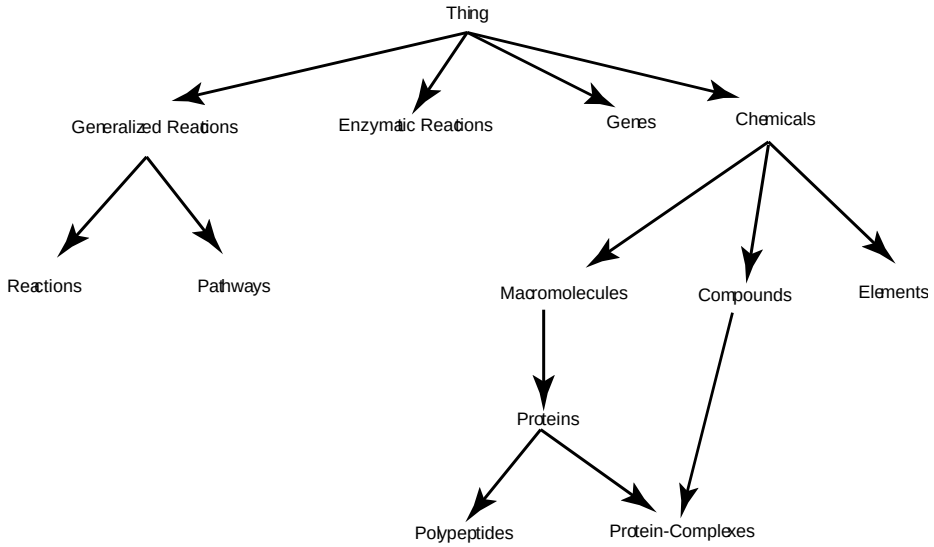

Figure 2.3: Top of the class hierarchy for the MetaCyc Knowledge Base (KB).

4. For studies of pathway evolution

5. As an aid in teaching biochemistry

MetaCyc aids in the process of metabolic engineering through genetic engineering which involves: [31]

- Inserting a new enzyme or pathway into an organism
- Replacing an existing enzyme or pathway with a substitute, or
- Removing an enzyme or pathway.

The installable application has a number of capabilities beyond those available through the BioCyc.org website: More query options, particularly on the full metabolic overview, such as

- Compare full metabolic maps of two or more organisms
- Find compounds, reactions, and genes on the overview
- Highlight enzymes controlled by a specified transcription factor

- Customizable partial gene maps
- Show an object in a species
- Programmable (Lisp and Perl-based APIs with documentation)

Paley, *et al.* [29] designed the pathway tools software as a reusable software tool for creating model-organism specific databases (MODs) such as EcoCyc for the species E.*coli k-12*. The pathway tools software [50] allows the EcoCyc database to be queried by providing a multitude of query operations and visualization tools. Both Ecocyc and MetaCyc are contained in a bigger collection of model organism databases called BioCyc. The majority of databases in the BioCyc collection are computationally derived databases that are generated by a program called PathoLogic. PathoLogic predicts the metabolic pathways of an organism from its genome; in that sense, the metabolic pathways in such databases are computationally derived, in contrast to the literature-derived pathways in the EcoCyc and MetaCyc databases. The input required by PathoLogic is an annotated genome for the organism, such as in the form of a Genbank entry. The output produced by PathoLogic is a new pathway/genome database for the organism. For example, the AgroCyc pathway/genome database for the bacterium *Agrobacterium tumefaciens* was created by the company SRI International using the PathoLogic program. In general, most of the metabolic pathways in a computationally derived database are predicted computationally, but in some cases, pathways that have been observed experimentally in the organism are added manually to the database. Additional unknown pathways are likely to be present in each organism. Hence, such databases have to be interpreted with caution by the life scientists. Chapter 5 provides a detailed discussion of the database design and the derived relational database schema for EcoCyc. Karp and Paley [27, 28, 32] provide more detailed descriptions of the capabilities and limitations of these tools.

2.1.3 BRENDA

BRENDA [65] is a primary collection of enzyme functional data. BRENDA is maintained and developed at the Institute of Biochemistry at the University of Cologne. Data on enzyme function are extracted directly from the literature by qualified scientists. Formal and consistency checks are automated by computer programs, each data set on a classified enzyme is checked manually by at least one biologist and one chemist.

2.1.4 EMP

The Enzymes and Metabolic Pathways (EMP) database [15] claims to be a unique and comprehensive electronic resource of biochemical data. EMP contains information that is indispensable in the analysis and mathematical simulation of metabolic pathways, reaction mechanisms, rate laws and a very wide spectrum of numeric data. The database is being constructed at Pushchino, Moscow region, Russia. It contains about 30,000 records derived from 15,000 original experimental journal publications.

2.1.5 WIT/ERGO

Overbeek, *et al.* [44] develop WIT (What Is There?), a WWW-based system to support the curating of function assignments made to genes and the development of metabolic models. It is described as 'an interactive metabolic reconstruction on the web'. It uses data from the EMP family of databases (see Section 2.1.4) and includes over 40 genomes as of year 2002. Its main purpose is to support comparative analysis of sequenced genomes and to generate metabolic reconstructions based on chromosomal sequences and metabolic modules from EMP. It also includes transport and signal transduction pathways.

The WIT web interface includes four components:

1. A functional overview that provides a hierarchical listing of pathways present in a chosen organism. But, this hierarchy is not consistent with hierarchies found in other databases.
2. Open reading frame pages that provide protein information and links to similar proteins.
3. Pathway pages that provide a list of proteins participating in a pathway in a particular species.
4. Assertion table that lists the presence or absence of a particular pathway member in each of the species.

2.1.6 ExPASy-biochemical pathways

The ExPASy (Expert Protein Analysis System) [17] proteomics server maintained by the Swiss Institute of Bioinformatics (SIB) is dedicated to the analysis of protein sequences and structures as well as 2-D poly acrylamide gel electrophoresis (PAGE). It provides a digitized version of the

EC Number (w)	Meaning
1	oxidoreductase
2	transferase
3	hydrolase
4	lyase
5	isomerase
6	ligase

Table 2.1: Top level classification of chemical reactions based on EC.

complete metabolic map being maintained by Roche Applied Science. This map includes the cellular and molecular processes and links to the ENZYME database (see Section 2.1.7).

2.1.7 ENZYME

ENZYME is a repository of enzyme nomenclature. ENZYME employs the recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (IUBMB) and contains those characterized enzymes having an EC (Enzyme Commission) number. The Enzyme Commission is an *ad hoc* committee formed in the 1950s to tackle the many difficulties arising from the uncontrolled naming of the rapidly increasing number of known enzymes. Some names were misleading, while others conveyed little or nothing about the nature of the reaction catalyzed. Enzymes catalyzing similar reactions sometimes had names suggesting they belong to different groups, while some enzymes of different types had been placed in the same group. For example, the pyrophosphorylases included both glycosyltransferases and phosphotransferases. In other cases, a name that had been well established for many years with a definite meaning, such as the term synthetase, was later employed with different meanings, thus causing confusion.

An EC number is of the form EC w.x.y.z where w, x, y, and z are integers. The main classification based on EC numbers is shown in Table 2.1. IUBMB [63] provides an official listing of EC numbers and their meanings. In particular, ENZYME provides EC number, recommended names, alternative names (if any), catalytic activity, cofactors (if any), pointers to the Swiss-PROT protein sequence entries(s) that correspond to the enzyme (if any), and pointers to human disease(s) associated with a deficiency of the enzyme (if any). Each entry in the ENZYME database is a bioreaction linked to SWISS-PROT sequences for enzymes that catalyze the reaction.

2.1.8 Summary

The databases described above are currently not complete and will not be complete in the near future. The KEGG database deserves first place with respect to user friendly access and the amount of data curated. Its nomenclature is consistent with international standards such as IUBMB and IUPAC. The design of the database and the classification of pathways, substrates, and reactions are in agreement with the expectations of the biologist. Hence, KEGG is the first choice for a typical user. For users interested in biochemical pathways of the E.*coli* bacterium, the EcoCyc system is the most comprehensive information resource on this model organism. The system requires a JavaScript-enabled web browser to provide direct links to the literature and the experimental data. With reference to the graphical representation of the metabolic maps, ExPASy provides more advanced and contextual information, such as subcellular location and interconnections between different cellular processes. Hence, ExPASy may be the choice of an advanced user. The individual metabolic pathways of KEGG are more consistent, clear, and simple to follow, though a complete metabolic map is not available. Some of these databases are commercializing and require license agreements for use. However, such license agreements are typically available free of cost to academic and nonprofit organizations. For example, MetaCyc and EMP require such license agreements. This is an indication of the potential demand for the information that these databases disseminate.

In summary, we note the following points about these databases.

1. Biological data are distributed worldwide and are mainly available as a web service or as downloadable flat files.
2. No standard nomenclature is followed for various biological entities.
3. The schema design and classification schemes of each database is different.
4. These databases are updated independently and often.
5. The user interface and manipulation tools for the pathways are significantly different in these databases.
6. Errors and inconsistencies are to be expected and accommodated (they are not perfect).

Table 2.2 summarizes quantitative information about the contents of these databases as of December 1, 2004.

Attribute	KEGG	MetaCyc	Brenda
Substrates	11323	3465	7668
Enzymes	4306	1665	83000
Reactions	5902	4873	4200
Genes	597295	1731	N/A
Species	174	240	9800
pathways	14409+237	496	

Table 2.2: The table gives a quantitative summary of KEGG, MetaCyc, and BRENDA databases as of December 1, 2004. KEGG clearly outnumbers MetaCyc and BRENDA on the substrate and reaction counts. However, BRENDA has the greatest number of distinct enzymes.

* Schomburg, *et al.* [55] provide summary for the BRENDA database. KEGG and MetaCyc summary was obtained partially from the downloaded database and partially from their respective websites.

2.2 Signaling Pathway Databases

2.2.1 CSNDB

The Cell Signaling Networks Database (CSNDB) is a database and knowledge-base for signaling pathways of human cells. It compiles information on biological molecules, sequences, structures, functions, and biological reactions that transfer the cellular signals. Signaling pathways are compiled as binary relationships of biomolecules and represented by graphs drawn automatically. CSNDB is constructed on top of ACEDB and the inference engine CLIPS, and has a linkage to TRANSFAC. Their final goal is to make a computerized model for various biological phenomena [60].

2.2.2 SPAD

The Signaling Pathway Database (SPAD) [56] is an integrated database for genetic information and signal transduction systems. A signal transduction pathway is a cascade of information transmission from the plasma membrane of a cell to its nucleus in response to an extracellular stimulus in living organisms. An extracellular signal molecule binds to a specific intracellular receptor and initiates the signaling pathway. There is a large amount of information in the literature about signaling pathways that control gene expression and cellular proliferation. At Kyushu University, Japan,

Hakozaki Hayashi-ku has developed an integrated database SPAD to study signal transduction mechanisms. SPAD is divided into four categories based on extracellular signal molecules, namely, growth factor, cytokine, hormone and stress, that initiate intracellular signaling pathways. SPAD contains information on interactions between protein and protein, and between protein and DNA, as well as DNA and protein sequences.

2.2.3 Drastic

The database resource for analysis of signal transduction in cells (DRASTIC) [38] is a collaboration between the Scottish Crop Research Institute (SCRI) and the University of Abertay Dundee. The web resource [61] is an effort to understand the molecular data from host-pathogen interactions.

2.2.4 Other databases

It should be noted that there are other biological databases that include, for example, protein-protein interaction data, and microarray gene expression data. For example, Biomolecular Relations in Information Transmission and Expression (BRITE) [10] is a database of binary relations for computation and comparison of graphs involving genes and proteins. It contains diverse sets of binary relations, including the generalized protein interactions that underlie the KEGG pathway diagrams, systematic experimental data on protein-protein interactions by yeast two-hybrid systems, sequence similarity relations by SEARCH, expression similarity relations by microarray gene expression profiles, and the cross-reference links between database entries.

Chapter 3

Review of Integration Initiatives

In Chapter 1, we argue that integration of useful and complementary biological databases is desperately needed. In this chapter, we review existing approaches and initiatives in biological data integration. Despite its practical importance, such integration is limited in practice. Since a vast majority of the early biological databases were nucleotide (genomic) and amino acid (proteomic) sequence databases, we first review some of the recent efforts towards integration of various sequence databases into a single non-redundant repository.

Notable initiatives in this direction include OWL, NRDB, and UniProt. The OWL database [11] is a repository for protein data. The OWL sequence database is a composite, non-redundant database assembled from a number of primary sources including translations of nucleic acid sequences. The highest priority is accorded to the SWISS-PROT data bank, with the addition of sequences extracted from NBRF/PIR (PIR1-3 only) and the Brookhaven PDB 3-dimensional structural database (NRL3D). Redundancy is avoided by comparison of sequences, with the elimination of exact duplicates and of sequences that differ only trivially. The data entries include references and other textual information, including cross-references to the PDB 3-dimensional structural and PRINTS databases.

NRDB is a non-redundant composite of the following sources: PDB sequences, SWISS-PROT, SWISS-PROTupdate, PIR, GenPept and GenPeptupdate. The database is thus similar in content to OWL, but contains more current information. However, strictly speaking, it is not non-redundant, but non-identical—meaning, only identical sequence copies are removed from the database. As a

result, NRDB is larger and less efficient to search than OWL.

UniProt [4] is an effort to combine the resources from PIR, SWISS-PROT and TrEMBL. UniProt is comprised of three components, each optimized for different uses. The UniProt Knowledgebase (UniProt) is the central access point for extensive curated protein information, including function, classification, and cross-reference. The UniProt NREF (Non-redundant Reference, UniRef) database facilitates merging of closely related sequences into a single record, thereby speeding sequence similarity searches. The UniProt Archive (UniParc) is a comprehensive repository that archives non-redundant protein sequences extracted from various public databases. Any information other than the sequences themselves should be retrieved from the respective databases using cross-references. We will see that keeping these cross-references alive all the time is very difficult as different databases are maintained, updated, and modified regularly.

Kuffner, *et al.* [34] analyze pathways in metabolic databases using differential metabolic display and integrated KEGG, Enzyme, and Brenda databases. For this, they represent reactions as Petri nets. Petri [49] introduced a special class of generalized graphs or nets that are now called Petri nets. The use of Petri nets leads to a mathematical description of the system structure that can then be investigated analytically. Figure 3.1 shows a Petri net representation of the reaction given by EC 1.11.1.10 taken from the BRENDA database. EC 1.11.1.10 refers to the chloride peroxidase reaction given by the equation 2 RH + 2 Cl- + H2O2 = 2 RCl + 2 H2O. In the Figure 3.1, X = Chlorine, Bromine, Iodine, but not Fluorine. An alkane is a saturated hydrocarbon such as methane, ethane or propane. The reactants (alkane, halogen, and peroxide) are shown on the left-hand side of the reaction, while the products (Alkyl halide and water) are shown on the right hand side. The enzyme Chloroperoxide with EC number 1.11.1.10 catalyzes this reaction by binding to the substrate.

Figure 3.2 shows a Venn diagram representation of matching substrate names from KEGG, BRENDA, and ENZYME databases. 2993 substrates were found common in all three databases, while 33 substrates were matched only between KEGG and BRENDA, 55 substrates were matched only between BRENDA and ENZYME, and 161 substrates matched only between KEGG and ENZYME. The large number (8953) of substrates in BRENDA remain unmatched with any of the substrates from either KEGG or ENZYME databases is a clear indication of the inconsistencies in naming substrates and the need for standardization. The PathMeld approach resulted in 82.6% perfect matches of substrates in Metacyc with corresponding substrates in KEGG. A perfect match is defined as either:

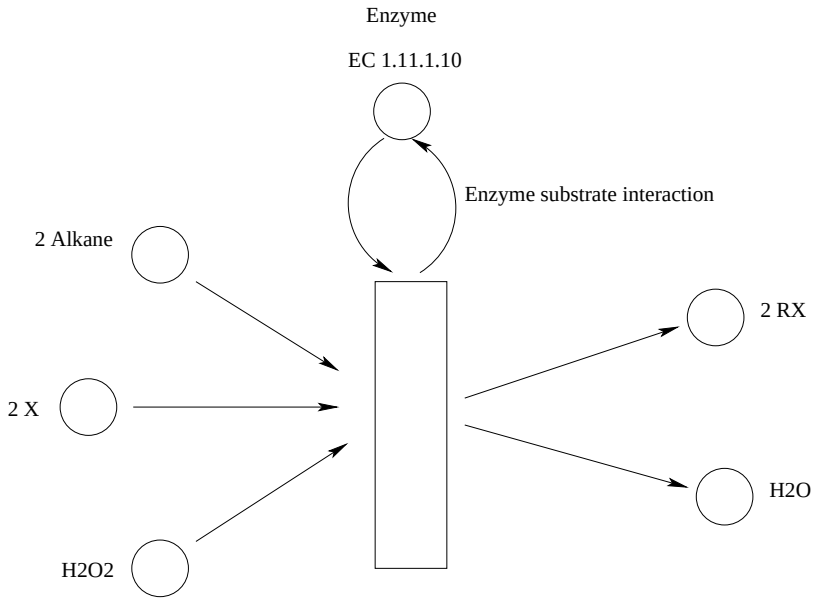

Figure 3.1: A Petri net representation of the reaction given by EC 1.11.1.10 (taken from [34]). EC 1.11.1.10 refers to the chloride peroxidase reaction given by the equation 2 RH + 2 Cl- + H2O2 = 2 RCl + 2 H2O. X = Chlorine, Bromine, Iodine, but not Fluorine. An alkane is a saturated hydrocarbon such as methane, ethane or propane. The reactants (alkane, halogen, and peroxide) are shown on the left-hand side of the reaction, while the products (Alkyl halide and water) are shown on the right hand side. The enzyme Chloroperoxide with EC number 1.11.1.10 catalyzes this reaction by binding to the substrate.

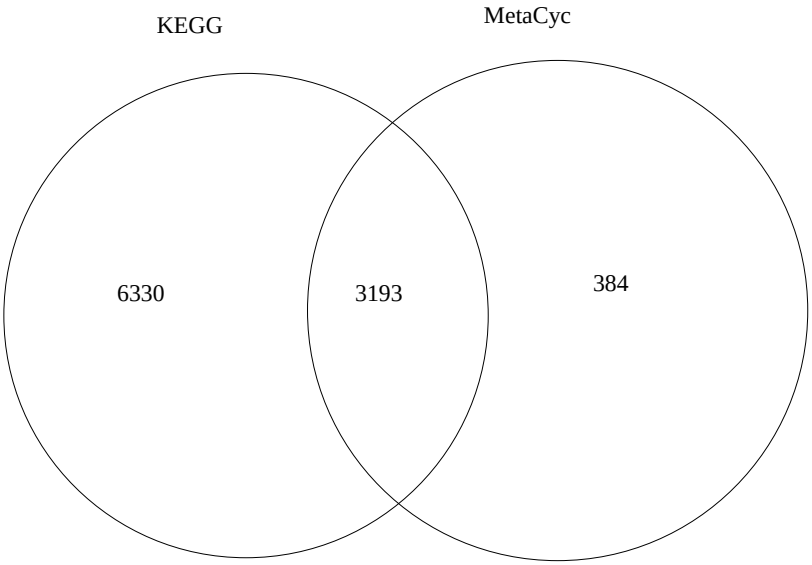

Figure 3.2: The results of substrate matching between the three databases KEGG, Brenda, and Enzyme (taken from Kuffner, *et al.* [34]). As we see, the total number of KEGG substrates is 3305, much less compared to the $11,323$ from Table 6.1. This is because, Kuffner, *et al.* [34] include only those KEGG substrates participating in what they consider main reactions. This indicates that majority of KEGG substrates are less important as they do not participate in the main reactions.

- An exact case insensitive match of the string containing one of the names of a MetaCyc substrate with one of the names of a KEGG substrate, or
- An exact case insensitive match of the string containing the chemical formula of MetaCyc with the string containing the chemical formula of a KEGG substrate.

By allowing errors to occur in a string match, we improved the perfect matching results by 3.83%. By further allowing close matches, we improve the results by 3.17%. A close match implies that the two substrate names under consideration are not exactly the same, but are closely related substrates. Examples of approximate matches based on each criteria are provided in Appendix D.

Note that a substrate may have several names but just one chemical formula. This may lead to the expectation that all substrates can be easily matched using the chemical formula. However, there are three reasons why this may be untrue.

1. The formats in which the chemical formula is stored may differ. For example, the KEGG

database stores the chemical formula for the substrate 2-nitrophenol in KEGG format as C6H5NO3. In contrast, the MetaCyc database stores the chemical formula for the same substrate 2-nitrophenol as (C 6)(H 5)(N 1)(O 3). In the PathMeld methodology, this issue is handled by a separate parser program that converts the chemical formula format of MetaCyc into the KEGG format. This formatted formula is used in the matching process.

2. A database may not have recorded the chemical formula even though it is well known. For example, the chemical formula for the substrate ATP (Adenosine Tri-Phosphate) is blank in MetaCyc but a chemical formula (C10H16N5O13P3) is provided in KEGG.

3. It is important to note the exceptions to the above definition of perfect match. Even when the chemical formula of MetaCyc and KEGG match perfectly, there may be cases where the two substrates are still not the same. For example, structural isomers share the same molecular formula but have different arrangements of atoms in the molecules. For example, both acetone (also called 2-propanone) and propionaldehyde (also called propanal) share the same molecular formula of C3H6O, but have different arrangements of atoms in the molecules. Structural isomers usually have different chemical and physical properties. Hence, it is important that substrate names are used for matching before adopting the chemical formula based matching.

Hence, it is advantageous if the database designers employ a standard format for chemical formula and make every effort to include one if available in the literature.

Lacroix [35] presents an approach to wrapping web data sources, databases, flat files, or data generated by tools through a database view mechanism. A wrapper performs two tasks:

1. Sends a query to a data source to retrieve data, and

2. Builds the expected output with respect to the virtual structure.

The approach involves wrappers that are composed of a retrieval component based on an intermediate object view mechanism, called "search views", mapping the source capabilities to attributes and an XML engine.

Chu, *et al.* [68] emphasize the need for database integration with the web for biologists to share data and information. They enumerate the current programming technologies such as CGI, ASP, JSP, Coldfusion, and PHP available for database integration with the web. Mishra, *et al.* [48] propose a system called Valis that utilizes exactly these tools. The architecture of the Valis system is shown

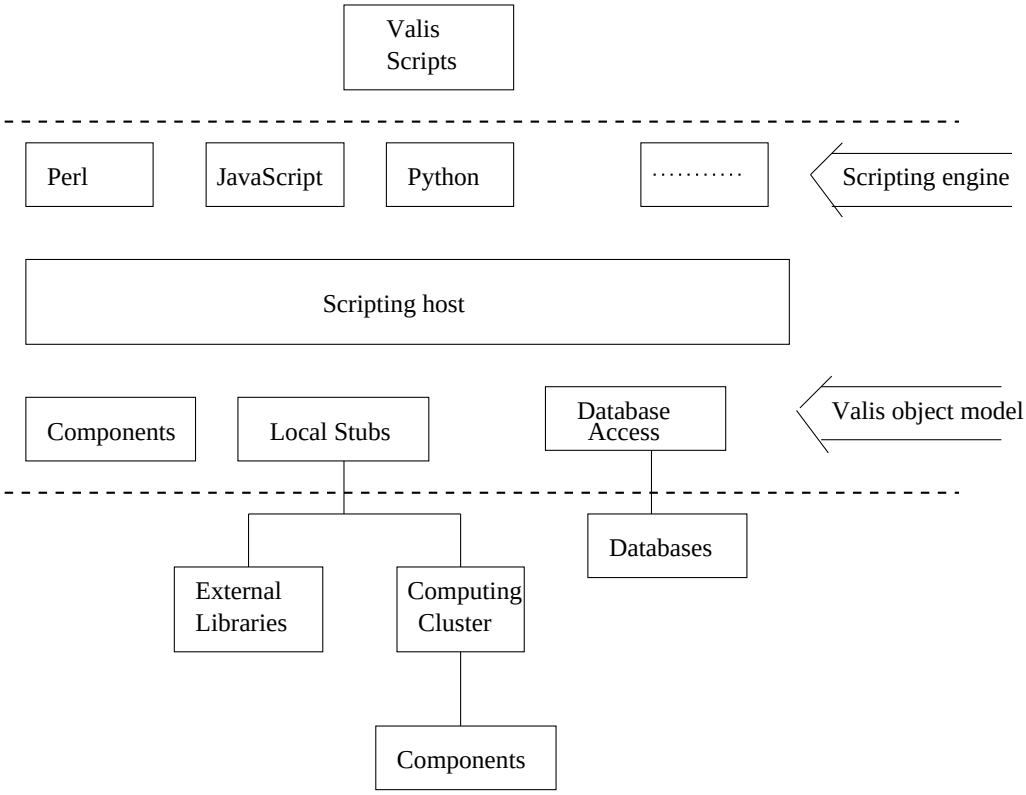

Figure 3.3: The Valis architecture.

in Figure 3.3. The problem with Valis is that it relies on high network bandwidth for retrieving query results; this could be an issue for large datasets. Hence, the performance of complicated queries on multiple data sources may be poor. Another problem is that the system needs to be updated very often since each of the web data sources is independently modified and updated or may even be disconnected. For example, several public databases such as ExPASy provide numerous hypertext links to the WIT and EMP websites. However, both web sources are unavailable for many months, yet the links to them from ExPASy remain.

A team at the Lawrence Berkeley National Laboratory [47] develop a somewhat related tool called "BioSig" utilizing Java, Enterprise Java Beans (EJB), and CORBA technologies. BioSig provides a data model for capturing experimental annotations and variables, computational techniques for summarizing large numbers of images, and a distributed architecture that facilitates distant collaboration.

One of the questions of interest to a biologist is: given a complete genomic sequence, is it possible to reconstruct a complete functional representation of the biological organism? What other contextual information is required? Considering the tremendous pace at which research is progressing in this direction, the answers to these questions may not be very far away. However, for this to be possible, easy access to complete biological data for a given organism is needed. A significant effort has been made in this direction in the case of Bacillus subtilis. Harwood and Moszer [19] sum up the decade long efforts in the study of Bacillus subtilis that has resulted in five different databases contributing complementing information:

1. The SubtiList database holds the core genomic data [42].

2. The Micado database contains phenotypic data generated in the framework of B. subtilis functional analysis programme [9, 52].

3. The Sub2D contains the results of analyzing two dimensional protein gels with the aim of deciphering new regulons and stimulons [7, 8].

4. The SubScript database organizes data generated by transcriptome analysis. The data is in accordance with the recommendations published by the Microarray Gene Expression Databases (MGED) consortium (Minimal Information About Microarray Experiments, MIAME) [12].

5. The SPID database contains information about two hybrid protein interactions [20].

Together, these databases contribute towards a comprehensive genomic resource for B. subtilis. A comprehensive resource will allow researchers to pose complex questions utilizing the knowledge distributed across the five component databases. An example query is as follows: Which genes located close to each other (SubtiList) are involved in sporulation (Micado), are co-activated in the presence of glucose rich medium (Sub2D and SubScript), and participate in a single protein complex (SPID)? Such queries demand a greater level of data integration. We show that the PathMeld methodology supports such queries in the metabolic pathway databases scenario.

XML is emerging as a standard for data exchange on the Internet. For biological pathway data specifically, the following three XML based standards are emerging:

1. Systems Biology Markup Language (SBML): Hucka, *et al.* [21] develop the XML based systems biology markup language. It is a model representation language for describing simulation models in systems biology. It focuses on representing the structure, parameters, and mathematical description of a biochemical pathway model.

2. Biochemical Pathways Exchange (BioPAX), [59] is an XML based biological pathways exchange format, namely, BioPAX. It focuses on molecule and interaction classification schemes and database cross-referencing for pathway components.

3. KEGG Markup Language (KGML): is not really a standard for pathways in general, but a data exchange format for the KEGG data alone.

Cuellar, *et al.* [13] develop CellML, an XML-based exchange format for defining and exchanging biological models. SBML is the closest relative of CellML. While SBML is meant to support basic biochemical network models, CellML covers a more general field of application, including electrophysiological and mechanical models as well as biochemical pathway models. Of the four standards, CellML is not very specific to biological pathways alone. Since KGML is not aimed at biological pathway data in general, the tie is between SBML and BioPAX. Unfortunately, there is no clear winner at this time. Automatic visualization and analysis tools are being developed for each of the two standard formats, namely, SBML and BioPAX. If both, a full model or pathway description and information about types of pathway components and database links are desired, BioPAX and SBML may be linked together. A hybrid XML document containing both BioPAX and SBML elements tied together using the CellML metadata standards could be the best solution at this time. However, deviating from these standard formats implies the overhead of developing custom visualization and

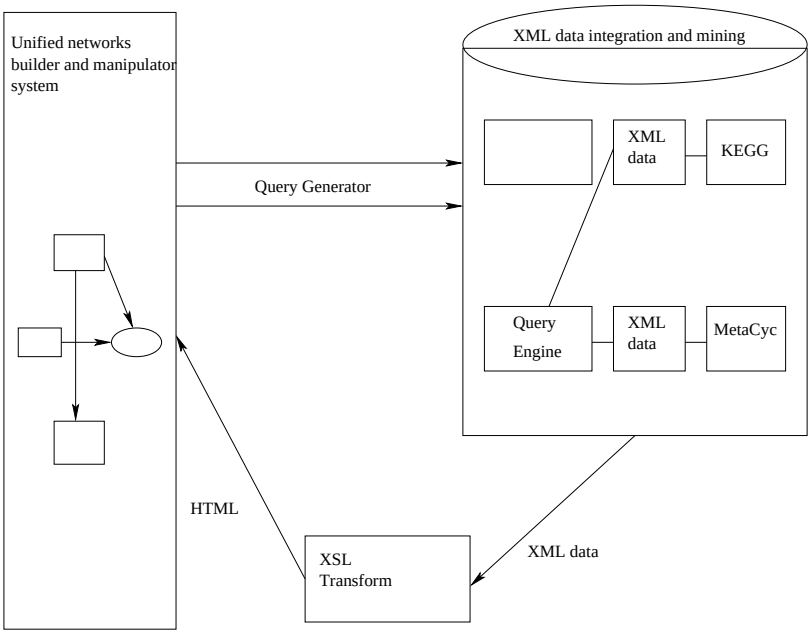

Figure 3.4: An XML architecture for integration and interoperation [41].

analysis tools for the hybrid format. Hence, until a single data standard emerges out of the two, the best option is to support both SBML and BioPAX. Since most individual databases support one of the two standards, this imposes the overhead of transforming between SBML and BioPAX to local formats, if web integration is desired.

Matoba, *et al.* [41] propose a prototype system for integration of heterogeneous biological XML data (see Figure 3.4). This system consists of three parts:

1. A Client (web browser) sends queries to the XML database system via a web server.

2. The XML database system stores biological data in XML format, an XML query engine, and meta knowledge. The query engine processes a demand from client and sends the resulting data to the XSL transform.

3. XSLT converts XML data to HTML and returns it to the client.

This assumes that the various data sources are readily available in a standard XML format. Currently, only KEGG database provides information in the KGML format. It requires the design and

implementation of an XML query engine that potentially is quite complex. This system (when developed) faces the same issues as those stated for the Valis design above. Currently, we can expect much support for executing complex queries directly available within database systems such as PostgreSQL. In addition, transforming the results in XML format to complex and dynamic pathway diagrams in HTML is a challenging task. This also requires that local XML schema are constantly updated. When most biological databases offer XML based web services, this approach may be a good choice.

Our PathMeld methodology overcomes most of these issues by implementing the unification at the database level. This is achieved by deciphering the relational database schema of PGDBs from their flat files. By analyzing individual PGDB schemas, we design a common relational database schema to represent metabolic pathway information. We suggest that this approach is more useful for serious data mining to gain useful biological insights, since we make all relevant data available for easy access locally.

One group of scientists are of the opinion that for complex problems such as biological data integration, experimental methods are mandatory. Macaulay, *et al.* [39] propose a model system to work with and experiment on, to better understand the real issues and challenges in the process. They envision an integrated database of genes such that, for a given gene in a given organism, the database should "horizontally" link sequence, structure, position and phenotype, and "vertically" link related elements of the same type that pertain to other genes in the same or in different organisms. They illustrate the approach by building as a model, an integrated database of human and mouse genes from Genbank and the human and mouse genome databases. It is interesting to note that they identified that 15% of the genes were apparently missing from the databases; $5to10\%$ of the genes had their links between sequence and genome databases missing; and $10to20\%$ of the entries classified as genes were misclassified. PathMeld can be viewed as a experimental model for integrating metabolic pathway databases and more than that, a methodology for integrating pathway genome databases.

Karp and Paley [45] develop a web server tool written in Common LISP, that allows existing graphical user interface applications written using the Common Lisp Interface Manager (CLIM) to hook easily into the WWW. Figure 3.5 shows the architecture of EcoCyc system and how the CWEST (Clim-WEbServer Tool) interfaces the various GUI applications with the EcoCyc system. InterPro [3] is an effort to integrate several complementing databases of protein structure motifs.

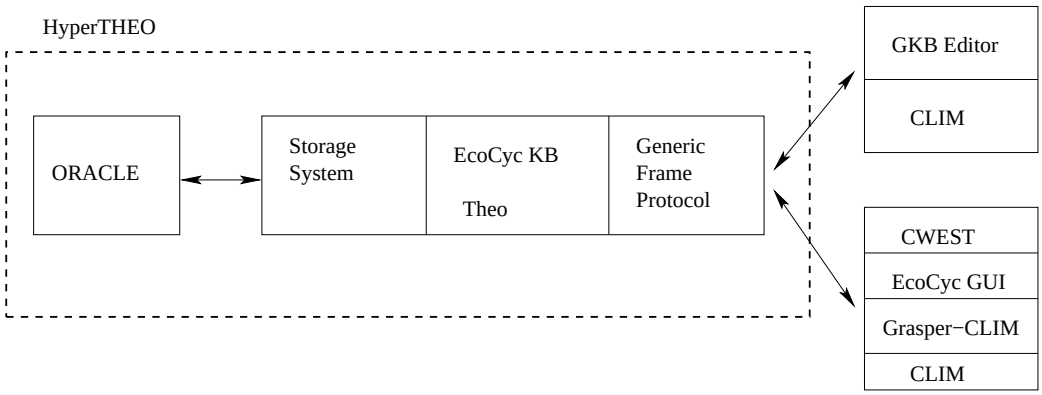

Figure 3.5: Architecture of the EcoCyc system.

Chapter 4

Research Objectives

The main objective of this research is to develop a general methodology for the integration of two PGDBs with their flat files as the starting point. For this purpose, KEGG and MetaCyc have been chosen as representative PGDBs. The detailed objectives of this study include:

1. Downloading and parsing the flat files of KEGG and MetaCyc.
2. Designing and implementing relational database schema amenable for population and integration of KEGG and MetaCyc. This includes filtering out unnecessary data from KEGG and MetaCyc and identifying the relationships among various entities and their attributes necessary for pathway integration (or merging).
3. Matching entities such as substrates, reactions, enzymes and pathways in KEGG and MetaCyc by comparing their attributes.
4. Developing heuristics for matching entities whose attributes do not have a perfect match.

One additional challenge here is to achieve these objectives in the light of their intended uses in building what are called "multi-modal networks" [51] to handle the complexity of biochemical pathways across organisms and tissue types. The database design must be flexible and conducive to achieve these goals.

A common problem in the analysis of biological information is the problem of multiple names for the same entity. As an example, acetyl coenzyme A is referred to in the literature and by biochemists interchangeably by any of the names given in table 4.1.

ACETYL-COA
ac-CoA
acetylcoenzyme-A
acetyl-S-CoA
ac-S-CoA

Table 4.1: Different names for acetyl coenzyme A.

ID	Common Name	Alias
C00268	BIOPTERIN	Dihydrobiopterin
C02953	BIOPTERIN	7,8-Dihydrobiopterin
C06313	BIOPTERIN	Biopterin

Table 4.2: Example of Biopterin from KEGG.

Even worse, in some cases, one name refers to different compounds (substrates). This is illustrated in Table 4.2 using the example of biopterin from the KEGG database. Parsing them computationally is a challenge since the parser must handle special characters, such as "," "-", digits and upper or lower case letters. In addition, there may be rearrangements in character order. An example from the MetaCyc database indicating different names of 5-Methyl-TetraHydroFolate (THF) is shown in Table 4.3. We found no entry for this compound in KEGG to compare to.

Substrate	Alias
5-METHYL-THF	N5-methyltetrahydropteroyl mono-L-glutamate
5-METHYL-THF	5-methyl-tetrahydrofolate
5-METHYL-THF	5-methyl-5,6,7,8-tetrahydrofolate
5-METHYL-THF	n5-methyltetrahydrofolate
5-METHYL-THF	CH3-THF
5-METHYL-THF	n5-CH3-THF
5-METHYL-THF	n5-methyl-THF
5-METHYL-THF	methyl-THF
5-METHYL-THF	methyl-tetrahydrofolate

Table 4.3: Example of different names of THF derived from MetaCyc.

Chapter 5

Research Methodology

This chapter presents a detailed description of the problems and issues involved in integrating PGDBs in general and KEGG and EcoCyc in particular. It then presents a detailed analysis of the EcoCyc and KEGG databases and tools, describing our methodology to integrate them in a biologically useful way. The chapter concludes with a comparison of the two PGDBs and the benefits of integration.

With KEGG and EcoCyc as the starting point, we review and evaluate the two PGDBs both computationally and biologically, distinguishing key features present or absent in each of them and the ways in which these features can be integrated or added and utilized computationally. Using the results of this review and evaluation, together with other available resources (see section on background), we came up with a local database design incorporating a multitude of information such as genome, pathway and signaling information.

5.1 Unification Strategy: PathMeld

In this section, we describe the PathMeld methodology in detail (refer to Figure 5.1). The overall approach for the unification of KEGG and MetaCyc is shown in the Figure 5.1: Some sample source code for performing each of the following steps is provided in the Appendices.

1. Download: Use the wget utility in UNIX to download the up-to-date flat file version of the MetaCyc database from the MetaCyc ftp site. This will put the flat file dumps of MetaCyc on to our local machine.

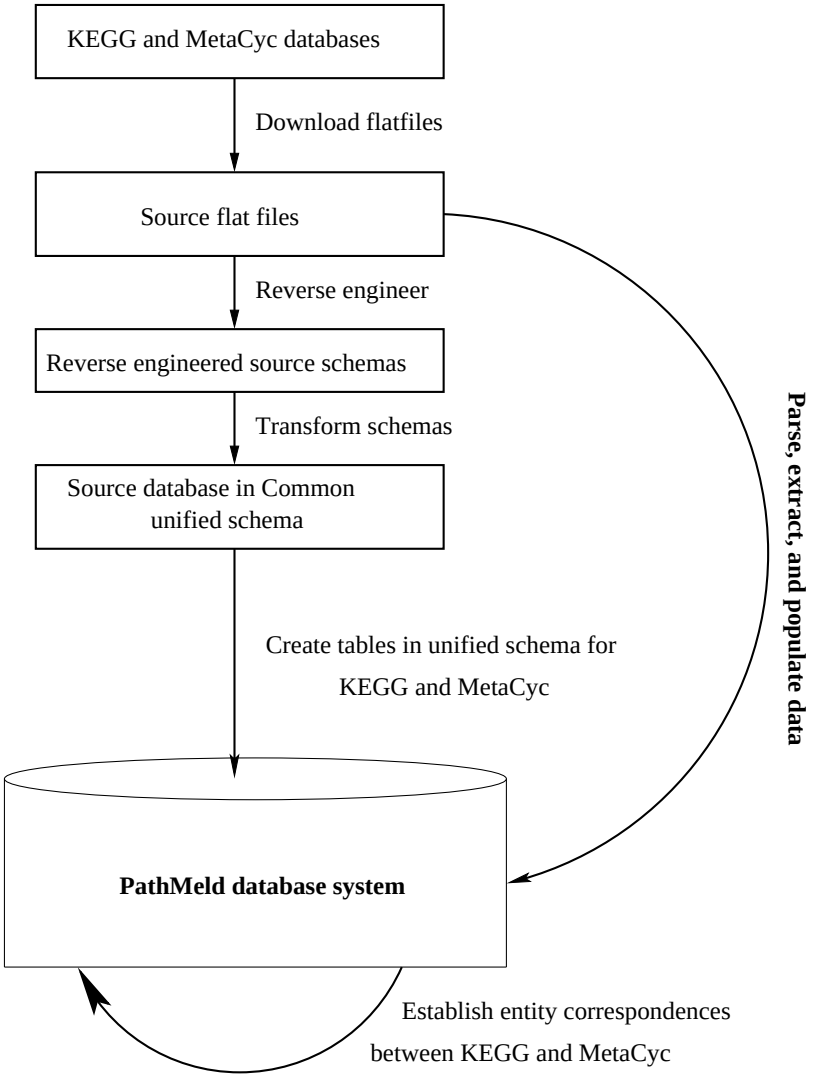

Figure 5.1: Diagramatic representation of the PathMeld unification methodology.

2. Analyze: The flat files thus downloaded are in different formats such as .dat (data file), .col (column separated file), .txt (text file), and .fasta (fasta format files). These files may contain data that are redundant or unnecessary for our purposes. In this case, our purpose is to obtain pathway information from both databases. Hence, a detailed analysis of these flat files is needed before proceeding. Refer to the Section 5.3 for details. This analysis will help us decide what information in the flat files can be eliminated from further consideration. The idea is to transform individual PGDB into a common unification schema shown in Figure 5.2.

3. Parse: Now that we have identified the required data fields from the MetaCyc flat files, we are ready to parse the flat files to extract the information we need and create a file in tabular format to input to the database table (refer to next steps). Refer to Appendix A for sample C code for parsing.

4. Create Database Tables: We create the database tables for the shortlisted fields that allow straightforward normalization. We utilize a PostgreSQL database for this purpose. Refer to appendix B for source code. Note that the primary key and foreign key constraints are not added during the creation of tables. Doing this will complicate the database population process due to data inconsistencies and flat file parsing. It is simpler to first populate the database tables without such constraints and to add constraints after data validation.

5. Populate the Database: Now, we have the data in tab delimited format and the database tables in PostgreSQL. Hence, this becomes a straightforward step of populating the database tables. The source code is provided in appendix.

6. Repeat 1-5: Repeat steps 1 through 5 above for the KEGG database.

7. Match the entities: We now have all the information we need about the compounds, reactions, enzymes and pathways from both KEGG and MetaCyc. We have to match the database entries for four entities, namely compounds, reactions, pathways and enzymes. Since we use Enzyme Commission (EC) numbers to establish enzyme and reaction correspondences across databases, it is a straightforward task. Hence, matching substrates between MetaCyc and KEGG forms the key to PathMeld methodology.

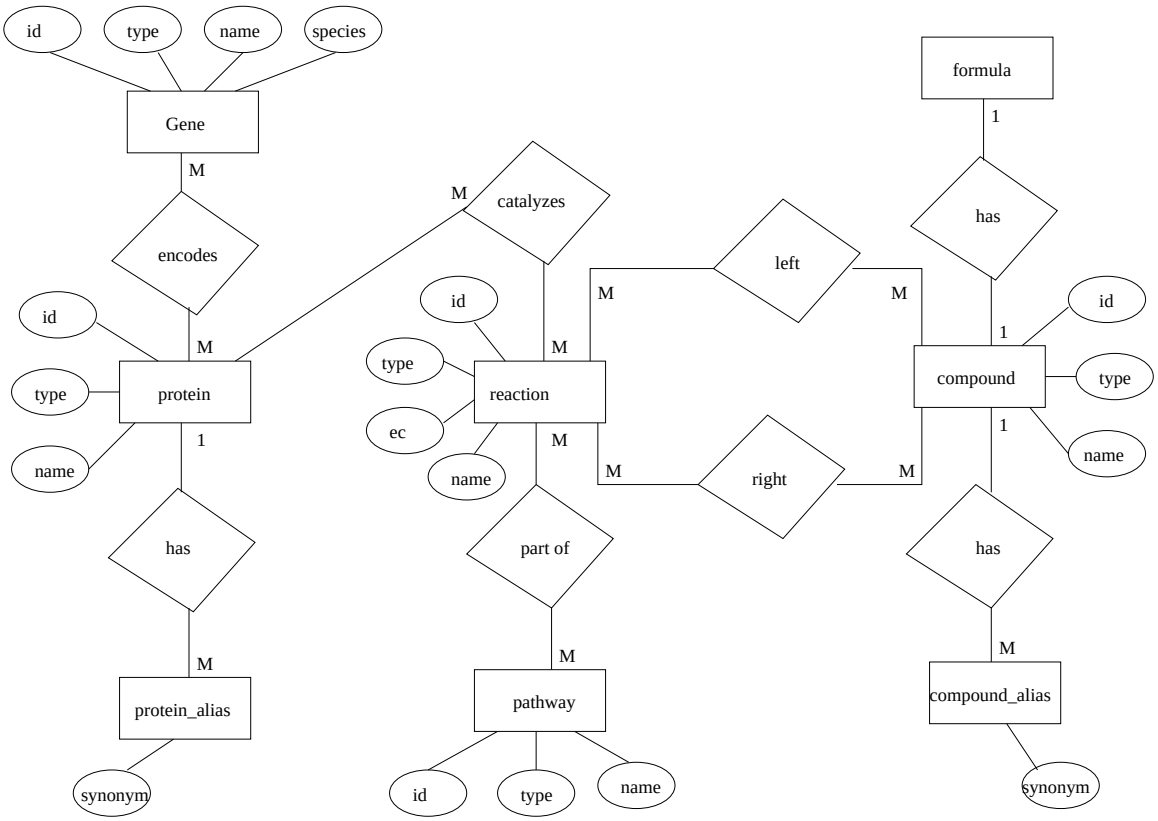

Figure 5.2: The entity relation diagram of the unification database schema that captures the common content of KEGG and MetaCyc databases. The schema has 8 entities, namely *gene*, *protein*, *protein_alias*, *reaction*, *pathway*, *compound_alias*, *compound*, and *formula*. Data on enzymes are subsumed in the *protein* entity. MetaCyc doesn't limit itself on enzymes alone hence the choice of the entity name.

5.2 Substrate Matching

As we saw in the previous section, substrate matching forms the key to PathMeld methodology. Here, we describe the methodology and techniques we used for this purpose. For substrates where these techniques fail, we identify why the failure occurs.

For matching (or to find if a match exists for) a substrate in one database to some substrate in the other, we need to compare different forms and properties of the substrates from the two databases. A heuristic we suggest in defining the aim of substrate matching is the following: Find the number of entries for substrates in the two databases. From Table 2.2, the number of distinct substrates in KEGG is about three times that of MetaCyc. Hence, we aim to match all the $3,465$ MetaCyc substrates with atleast one of the $11,323$ KEGG substrates. Ideally, we would like all MetaCyc compounds to be matched to one or more substrates in KEGG. But, this may not happen due to the issues described in Section 6.2. We handle most of these issues using the techniques described in this Section.

If we find an exact match for a compound in MetaCyc with a compound in KEGG, they can be assumed to be participating in same reactions. We say an exact substrate match occurs when:

1. A MetaCyc name or chemical formula matches exactly (string comparison) with the corresponding attribute of a KEGG substrate. This is achieved by a direct string comparison within the PathMeld postgres system using SQL queries.

2. The MetaCyc name or formula is a reordered case of the KEGG name or formula. This is achieved by a stored C subroutine that compares the character counts of each character from the two strings.

3. A MetaCyc name differs from a KEGG name in a subtle way but are adjudged same. This is accomplished by manual analysis of the agrep results.

A close substrate match occurs when a MetaCyc name closely or approximately matches with a KEGG name. Such substrates are identified as closely related substrates and should not be interpreted the same, in strict sense.

We present a heuristic driven approach for one-to-one mapping of the substrates between KEGG and MetaCyc. We subject the MetaCyc names and formula to pre-processing. The substrate names in the MetaCyc database contain html tags and non-characters such as <sub>, <sup>, <i>, <l>,

&, and $. The MetaCyc chemical formula are stored in lisp format in the database while KEGG stores it as a continuous string. Hence, we subject MetaCyc chemical formulae to transformation into KEGG format to make them directly comparable. After pre-processing, we first match the MetaCyc substrates with KEGG substrates by comparing their names and synonyms. Second, we match by their chemical formula. For all substrates that remain unmatched at this point, we look for reordered names, synonyms, or chemical formula match (same string length and characters but in different order).

While majority of the MetaCyc substrates stand matched at this stage (83%), we need to understand the inconsistencies and errors in the naming of the remaining 17% of the MetaCyc substrates that are not matched. For this, we use the approximate string matching of the substrate names and synonyms using the AGREP utility to string compare each name of the unmatched MetaCyc substrates with each name of all the KEGG substrates. Further, we apply transitive property on all names of a substrate that match with one or more of its names. Suppose a MetaCyc substrate x has names A, B, and C, and substrate y has names M and N. agrep matching will give us the results of comparing A and M, A and N, B and M, B and N, C and M, and C and N. Now, if we judge that the comparison B with N is a correct match, meaning substrate x and substrate y are the same, then by transitivity, we say, A, B, and C all match with both M and N. The results of this technique on KEGG and MetaCyc compounds are presented in the chapter 6.

5.3 Reverse Engineering of PGDBs

5.3.1 EcoCyc

By analyzing the fourteen *.dat files, we can derive the following fourteen attribute-value tables and their inter-relations for EcoCyc as shown in Table 5.1.

* Refer to E for meanings of biological terms used in this table.

From the EcoCyc PGDB in flat file format, we can ignore all the files with .col extension (tab delimited files) except the proteincplx.col file in constructing the DB schema as they form a subset of the *.dat files. The database relations and attributes needed are dictated by the designed unified schema given in Figure 5.2. Hence, we ignore the following files for the reasons given below:

- enzymes.col, genes.col, pathways.col, protcplxs.col, and transporters.col, are tab delimited

Relation*	Primary key	Foreign keys	Comment
BindingRxns	bindrxnid	promoterid (reactant), proteinid (activator), proteincplxid (activator), dnabsid (activator or reactant)	references four tables
Class	clsid	none	lists all classes of compounds and is referenced by other tables
Compounds	compoundid	bindrxnid (appears-in)	Required entity
DNABinding Sites	dnabsid	tuid (component-of), promoterid (regulated promoter), bindrxnid (appears-in)	
Enzymatic Reactions	enzrxnid	compoundid (activator, cofactor, inhibitor), proteinid (enzyme)	Required entity
Genes	geneid	proteinid (product)	Required entity
Pathways	pwyid	clsid (primaries), compoundid (primaries, pwy-links), rxnid (rxn-lst, predecessor), proteinid (primaries), enzrxnid (enzyme use)	pathway table is never referenced from any other table
Promoters	promoterid	tuid (component-of), bindrxnid (appears-in)	
Proteins	proteinid	geneid (from), bindrxnid (appears-in)	Required entity
Publications	pubid		
Reactions	rxnid	promoterid (reactant), compoundid (left, right, reqts), clsid (left, right)	Required entity
Regulons	regulonid	proteinid (component-of, components), geneid (gene), bindrxnid (appears-in)	not required
Terminators	terminatorid	tuid (component-of)	references table transunits only
Transunits	tuid	terminatorid (components), promoterid (components), dnabsid (components)	Not required

Table 5.1: The table shows the various relations derived from the EcoCyc flat file database. The relations that are required for pathway reconstruction are indicated in the Comment column.

versions of their .dat counterparts.

- ecobase.ocelot summarizes the data in the entire EcoCyc database and is covered by the individual files described in Table 5.1.
- pubs.dat lists publication references with authors and title, not required for constructing pathways.
- protseq.fasta contains the amino acid sequences of the individual proteins included in the proteins relation of the EcoCyc database, not required for our purposes.

5.3.2 MetaCyc

The MetaCyc database in flat file format as of February 14, 2003 18:58:48 contains a subset of the file names and relations in EcoCyc descirbed in Table 5.1. MetaCyc can be viewed as a superset of EcoCyc in the sense that MetaCyc includes pathway information of several other species in addition to*E. coli*. However, MetaCyc database does not include the following files and relations given in Table 5.1 such as publications, transunits, regulons, promoters, DNA binding site information, binding reactions, and terminators. In addition, we also ignore the files described in Section 5.3.1. Now considering the filtered entities, we need to further filter several attributes of each of these entities (compounds, genes, reactions, enzymatic reactions, proteins, and pathways) that do not serve the objectives. The various entities and their attributes included or excluded from consideration are listed below:

1. For the COMPOUND entity, the following attributes are considered for inclusion in the unified schema:

   ```
   UNIQUE-ID
   COMMON-NAME
   CHEMICAL-FORMULA
   SYNONYMS
   ```

   while the following attributes are excluded:

   ```
   TYPES and
   COMMENT
   ```

   For example, an entry for the compound 3 keto adipyl CoA from the flat file is as follows:

```
UNIQUE ID - 3 KETO ADIPYL COA
TYPES - 3 KETOACYL COA
COMMON-NAME - 3 ketoadipyl CoA
CHEMICAL FORMULA - (C 27)
CHEMICAL FORMULA - (H 42)
CHEMICAL FORMULA - (N 7)
CHEMICAL FORMULA - (O 20)
CHEMICAL FORMULA - (P 3)
CHEMICAL FORMULA - (S 1)
SYNONYMS - 3 keto adipyl coa
SYNONYMS - 3 oxoadipyl CoA
```

Note the chemical formula format of the MetaCyc data. The chemical formula is provided as a string in the KEGG database. For 3 keto adipyl CoA, the formula from KEGG database format will be: C27H42N7O20P3S.

2. For the REACTION entity, the following attributes are included in the unified schema:

```
UNIQUE-ID
TYPES
COMMON-NAME
EC-NUMBER
LEFT
RIGHT
```

The following attributes of the REACTION entity are excluded:

```
COMMENT
EC-LIST
ENZYMATIC-REACTION
IN-PATHWAY
SYNONYMS
```

Note that some of these excluded attributes such as IN-PATHWAY (indicating the pathways in which the reaction participates) may be required attributes. However, due to data redundancy, such attributes occur in multiple flat files. Hence, such redundant data will be considered only once with reference to the most relevant entity, in this case, IN-PATHWAY is the same as REACTION-LIST attribute in the PATHWAY entity. A similar relation exists between REACTION and ENZYMATIC REACTION entities in the MetaCyc database and is explained in Section 6.2.3.

3. From the enzrxns.dat file, the following attributes of the enzymatic reactions are included in the unified schema:

```
UNIQUE-ID
COMMON-NAME
REACTION
ENZYME
```

and the following attributes are excluded:

```
REACTION-DIRECTION
ACTIVATORS
ACTIVATORS-ALLOSTERIC
ACTIVATORS-MECHNOTSTATED
ACTIVATORS-NONALLOSTERIC
ALTERNATIVE-COFACTORS
ALTERNATIVE-SUBSTRATES
INHIBITORS
INHIBITORS-ALLOSTERIC
INHIBITORS-COMPETITIVE
INHIBITORS-MECHNOTSTATED
INHIBITORS-NEITHER
SYNONYMS
```

We do not require information about activators, inhibitors, cofactors, and synonyms for the enzymatic reaction names at this point and are hence excluded.

4. From the proteins.dat file, the following attributes of the protein entity are included in the unified schema:

```
UNIQUE ID
TYPES
COMMON-NAME
GENE
SYNONYMS
SPECIES
```

If we are interested in a table of enzymes, we can do that by simply using only those records from the protein database that contain an entry for the field CATALYZES. Alternatively, the same can also be achieved using the ENZYME attribute in the enzrxns.dat file. Hence, the attribute CATALYZES is excluded here. We exclude the following attributes of protein entity, some because they are not needed and others because the data is not available at this point.

```
COMMENT
COMPONENT OF
COMPONENTS
INSTANCE NAME TEMPLATE
LOCATIONS
CATALYZES
```

5. From the pathways.dat file, the following attributes of the pathway entity are included in the unified schema:

   ```
   UNIQUE-ID
   COMMON NAME
   REACTION LIST
   ENZYME USE(M) - (RXNID ENZRXN_LIST), (RXNID ENZRXN_LIST)
   TYPES - Pathways
   ```

   while the following are excluded:

   ```
   IN-PATHWAY(M) - PWY1G-158
   SUPER-PATHWAYS(M) - PWY1G-158
   SYNONYMS(M) - MSH metabolism
   SPECIES(M) - MTBRV
   ```

   Information about pathways within pathways are not required for our purposes and are excluded.

6. From the genes.dat file, the following attributes of the gene entity are included in the unified schema:

   ```
   UNIQUE-ID
   COMMON-NAME
   TYPES
   ```

   and the following are excluded:

   ```
   PRODUCT
   PRODUCT-TYPES
   COMMENT
   ```

   Genes encode gene products. We are interested in gene products of type protein only at this point and hence, we do not include all gene products here. Instead, we include a link to gene from the protein table using the information herein.

5.3.3 KEGG

From the flat file download of the KEGG system, the ligand database contains all the required data files. A description of each file in the ligand directory of release 26.0 is given below.

1. compound: COMPOUND section of the LIGAND database.
2. ECtable: Enzyme EC number classification table.
3. enzyme: Enzyme section of the LIGAND database.
4. ligand.txt: User manual of the LIGAND database.
5. ligand.doc: The ligand database user manual in Word format.
6. ligand.weekly.last.tar.Z: compressed file of the ligand database updated weekly.
7. reaction: REACTION section of the LIGAND database.
8. reaction_name.lst: List of reactions extracted from the DEFINITION field of the REACTION section.
9. reaction.lst: List of reactions extracted from the EQUATION field of the REACTION section.
10. reaction_main.lst: Almost the same as reaction.lst except that the compounds in each reaction are limited to the chemical compounds shown in the KEGG pathway diagrams, i.e. possibly main reactants, and the information on the direction.
11. README: Describes the contents of the files in ligand directory.
12. gif.tar.Z: GIF format compound structure data. By uncompress and tar xvf commands, the files C?????.gif are created in the gif directory. C????? is the compound ID.
13. merged_compound.lst: List of merged compound with the first column for the obsolete compound ID and the second coloumn for the currently used compound ID.
14. mol.tar.Z: MDL mol file format compound structure data. By uncompress and tar xvf commands, the files C?????.mol are created in the mol directory. C????? is the compound ID.

Chapter 6

Results

This chapter presents in detail the unification issues identified, detailed comparison of the KEGG and MetaCyc databases, results of integration and the PathMeld methodology applied to KEGG and MetaCyc. The PathMeld methodology unifies KEGG and MetaCyc flat file databases by establishing substrate correspondences between them. The resulting PathMeld database system is organized into three schemas, namely, kegg, metacyc, and unify. The unify schema stores the KEGG and MetaCyc substrate matches and their match type.

6.1 Results of Substrate Matching

We report the results of substrate matching in this section. Since we use Enzyme Commission (EC) numbers to establish enzyme and reaction correspondences between databases, matching substrates between MetaCyc and KEGG forms the key to PathMeld methodology. From Table 2.2, the number of KEGG substrates is about three times that of MetaCyc, we aim to match each MetaCyc substrate with one or more of KEGG substrates. The chapter concludes with a case study of the Glycolysis pathway from the KEGG and MetaCyc databases.

Table 6.1 reports the results of substrate matching. Table 6.2 reports the results of substrate matching using the agrep tool in tabular form. 82.6% of the substrates in MetaCyc were matched accurately to corresponding substrates in KEGG. To investigate how many of the remaining 17.4% substrates are indeed absent from KEGG, we employ an approximate string matching tool called

	Exact Name matches	Exact Chemical Formula Matches	Total Inputs to agrep	Total
Distinct MetaCyc Substrate IDs	2342	608	515	3465
Distinct MetaCyc substrate Names	6052	1183	842	8077
Distinct KEGG Substrate IDs	2367	2383	11323	11323
Distinct KEGG substrate Names	5453	8477	17589	17589

Table 6.1: Table summarizing the results of exact substrate name and chemical formula matching. The column "Inputs to agrep" indicates the KEGG and MetaCyc substrates that need further analysis to understand why they fail exact substrate match criteria.

agrep. By applying the transitivity property on the agrep results as described in Section 5.2, the results of substrate matching improved by 3.83% when correct matches from agrep are included and by 7% when even close matches are included. We identify that the 7.45% of MetaCyc substrates that are classified as false matches by agrep and 3.68% that are not matchable with any KEGG substrates are likely to be absent from KEGG. Applying pre-processing steps improved substrate matching by 2%. Appendix D.3 lists all MetaCyc substrate names that do not exist in the KEGG database.

Figure 6.1 shows the results of substrate matching from MetaCyc to KEGG. The number at the intersection of KEGG and MetaCyc shows the number of distinct MetaCyc substrates matched with one or more of KEGG substrates based on either exact name, exact chemical formula, or approximate name match. This includes the close matches from the agrep results. The total MetaCyc substrate matches amounts to 88.87%.

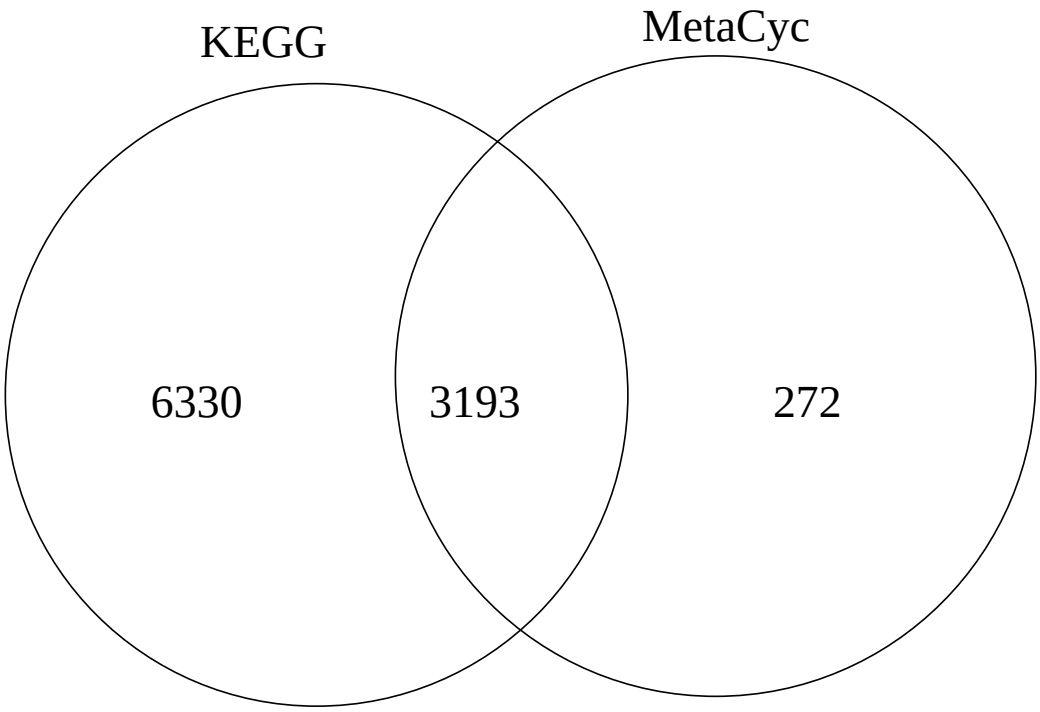

Figure 6.1: A Venn diagram depicting the results of substrate matches from MetaCyc to KEGG in December 2004. The number at the intersection of KEGG and MetaCyc shows the number of distinct MetaCyc substrates matched with one or more of KEGG substrates based on either exact name, exact chemical formula, or approximate name matches.

	Total Inputs to agrep	Exact match	agrep Close match	agrep Negative match	Remain Unmatched
Distinct MetaCyc Substrate IDs	515	132	111	257	127
Distinct MetaCyc substrate Names	842	166	149	367	159

Table 6.2: Table summarizing the results of approximate substrate matching using the agrep utility on substrate names. The table shows the classification of the MetaCyc substrates that remain unmatched from Table 6.1. This, along with Table 6.1 completes the classification of all the 3465 MetaCyc substrates into mutually exclusive groups based on their match criteria and status.

6.2 Unification issues identified

Many bioinformatics applications developed in recent years, especially pathway databases, illustrate the importance of DB content to solving computational problems [26]. For example, prediction of pathways of an organism from its genome requires an accurate and well-designed pathway DB. Currently, there are hundreds of specialized biological databases throughout the world that provide the biologists with a multitude of information that may be of use in advancing biological research [6]. Zimmer, *et al.* [34] propose a differential metabolic display technique for examining the differences between different databases by using Petri net representations of the biochemical pathways from different databases. Some of the issues involved in pathway reconstruction using genomic/gene expression data as the starting point are the following. The first one stems from the use of EC numbers for mapping genes to metabolic pathway diagrams [43]. In addition to the information on sub-cellular location, the information on tissue specificity (enzymes acting on a specific tissue) should also be considered for organisms with a large number of genes, such as mouse and human.

6.2.1 Subcellular location

Sub-cellular locations include the mitochondrion, cytoplasm, chloroplast, nucleus, vacuoles, endoplasmic reticulum, golgi complex, peroxisomes, and glyoxisomes. Tissue specificity in plants might

refer to leaves, roots, stem, or flowers. More specifically, enzymes may act in the following tissues: vascular tissue (xylem, phloem), ground tissue (parenchyma, colenchyma), or epidermal tissue. A database schema that represents information about sub-cellular locations and tissue specificity of each enzyme from literature and existing databases will support more accurate pathway reconstruction from gene expression data. One of the areas in which pathway reconstruction plays a vital role is in creating new analysis algorithms for extracting new insights from pathway networks to aid drug design by analyzing diseased human pathway networks or by predicting optimal drug target for antimicrobial drug design.

6.2.2 Limitations of EC numbers: Ortholog Identifiers

In KEGG database, EC-number refers to the enzyme. In MetaCyc database, EC-number refers to a reaction. Some reactions are catalyzed by a complex unit of several enzyme subunits (or orthologs) making it hard to assign EC numbers to these individual subunits. In such cases, an EC number may be assigned to the complex as a whole and not to one particular subunit within the complex. This makes it difficult to make a one-to-one correspondence from KEGG to MetaCyc or vice versa. We explain this with an example of chalcone isomerase below:

Consider the EC-NUMBER - 5.5.1.6. In KEGG, it refers to the enzyme chalcone isomerase. Reactions R02446 (4',5,7-Trihydroxyflavanone lyase (decyclizing)) and R02898 (Flavanone lyase (decyclizing)) are catalyzed by this enzyme. In MetaCyc, there are three reaction IDs with EC-NUMBER 5.5.1.6 given by APIGNAR-RXN, CHALFLAV-RXN, and CHALCONE-ISOMERASE-RXN. These three reactions are all identified by the same common name, Chalcone Isomerase, and same EC-number 5.5.1.6. However, they have different reactants and products. Table 6.3 illustrates the four different enzymatic reactions and the four different enzymes that catalyze them. These four ezymatic reactions are referred to by the same reaction APIGNAR-RXN with the common name "chalcone isomerase" but are catalyzed by different enzymes. Notice however that, the other two MetaCyc reaction IDs, CHALFLAV-RXN and CHALCONE-ISOMERASE-RXN do not reference any enzymatic reaction. Also, chalcone isomerase refers to an enzyme in the KEGG database while it refers to a reaction in MetaCyc. This inconsistency questions the use of EC-numbers as a standard in establishing reaction and enzyme correspondences across databases.

All these four enzymatic reactions are basically called by the same common name "chalcone

Enzymatic Reaction	Enzyme	Reaction
ENZRXN1F-771	AT1G53520-MONOMER	APIGNAR-RXN
ENZRXN1F-1393	AT3G55120-MONOMER	APIGNAR-RXN
ENZRXN1F-2434	AT5G05270-MONOMER	APIGNAR-RXN
ENZRXN1F-2453	AT5G66220-MONOMER	APIGNAR-RXN

Table 6.3: Different enzymatic reactions referred to by the same reaction ID APIGNAR-RXN in MetaCyc.

isomerase" but are catalyzed by different enzymes. Notice however that, the other two reactions (CHALFLAV-RXN and CHALCONE-ISOMERASE-RXN) referred to by the same EC-number are not enzymatic. Also, chalcone isomerase refers to an enzyme in the KEGG database while it refers to a reaction in MetaCyc. Due to this inconsistency, the EC-number cannot really be directly considered as a standard in the unification of such databases.

6.2.3 Relationship between reaction and enzymatic reaction

A record in RXN can reference multiple records in ENZRXN and vice versa. This is because of the Many-Many relationship between Enzyme and Reaction. One reaction can be catalyzed by many enzymes. One enzyme can catalyze multiple reactions. Example: Consider the reaction - Chalcone isomerase It references 4 enzymatic reactions as shown in Table 6.3.

```
UNIQUE-ID - APIGNAR-RXN
COMMON-NAME - Chalcone isomerase
EC-NUMBER - 5.5.1.6
ENZYMATIC-REACTION - ENZRXN1F-771
ENZYMATIC-REACTION - ENZRXN1F-1393
ENZYMATIC-REACTION - ENZRXN1F-2434
ENZYMATIC-REACTION - ENZRXN1F-2453
LEFT - APIGENIN
RIGHT - NARINGENIN-CMPD
SYNONYMS - Chalcone--flavonone isomerase
```

This reaction record APIGNAR-RXN references the following four records of enzymatic reaction:

1. UNIQUE-ID - ENZRXN1F-771
   TYPES - Enzymatic-Reactions
   COMMON-NAME - chalcone isomerase
   ENZYME - AT1G53520-MONOMER
   REACTION - APIGNAR-RXN

2. UNIQUE-ID - ENZRXN1F-1393
   TYPES - Enzymatic-Reactions
   COMMON-NAME - chalcone isomerase
   ENZYME - AT3G55120-MONOMER
   REACTION - APIGNAR-RXN

3. UNIQUE-ID - ENZRXN1F-2434
   TYPES - Enzymatic-Reactions
   COMMON-NAME - chalcone isomerase
   ENZYME - AT5G05270-MONOMER
   REACTION - APIGNAR-RXN

4. UNIQUE-ID - ENZRXN1F-2453
   TYPES - Enzymatic-Reactions
   COMMON-NAME - chalcone isomerase
   ENZYME - AT5G66220-MONOMER
   REACTION - APIGNAR-RXN

Note that the attribute that differentiates the four enzymatic reactions is the particular enzyme that catalyzes the reaction.

6.3 Implementation Issues

In this section, we outline the issues handled during the implementation of the proposed methodology.

- Parsing only one flat file to populate one database table is not sufficient.

  While parsing the MetaCyc flat files to collect data for the protein table, it is not enough to just parse the proteins.dat file. Some gene products are listed as proteins in the genes.dat file but are not present in the proteins.dat file. Similarly, while populating the compound-left-of-reaction and compound-right-of-reaction tables, we need to parse both the compounds.dat and proteins.dat files since they both can participate as reactants or products in reactions. Hence, it should be noted that it is not enough to just parse a single flat file for collecting complete information for a database table. All relevant tables should be scanned for data and distinct records be loaded into the database.

- Intermediate database tables are needed.

  For example, the compound chemical formula are presented in different formats in KEGG and MetaCyc databases as described in Section 5.3.2. Hence an intermediate table is used to convert the MetaCyc format to KEGG format.

- Transformation of substrate names and chemical formula from one of the component databases is required for comparison.

  In order to achieve accurate and complete substrate matching, transformation is needed on one of the component database entries so that we have inputs in a comparable format. Here, we transform MetaCyc names to remove all unnecessary and extraneous characters and html tags described in Section 5.1

- A biochemist intervention is needed in evaluating the approximate substrate matches from agrep.

  The PathMeld methodology is automated to a certain extent. Steps 1 to 6 of the PathMeld methodology given in Chapter 5 completes in one flow. This means, the unfication process with 82.6% of the substrate matching is automated. However, if approximate matching of the remaining 17.4% substrates is desired, a biochemist intervention is needed to judge the correctness of the approximate match results from agrep. Once the classification of the agrep results into correct, close, incorrect, or no match is complete, the remainder of the methodology is again automated.

6.4 Comparison of MetaCyc and KEGG

1. MetaCyc supports the Biological Pathways Exchange (BioPAX) standard. KEGG develops its own standard namely, KGML (see Chapter 3).

2. MetaCyc contains extensive comments about enzymes and pathways that KEGG lacks [33].

3. MetaCyc cites the primary literature sources based on which the pathways and enzyme data were obtained. KEGG has no literature citations [33].

4. KEGG pathways are usually larger than their MetaCyc counterparts. This is because, KEGG combines together a number of related pathways from several different species in one pathway

diagram. MetaCyc super-pathways perform similar function allowing users to view interconnections among multiple pathways [33]. Example: Glycolysis pathways in KEGG and MetaCyc (a detailed treatment is provided in section 6.5).

5. MetaCyc contains the different pathway variants observed in different species. KEGG does not explicitly record such pathway variants [33].

6. KEGG does not contain information about which pathways are observed in which species. However, KEGG does allow users to easily view, for a given pathway, which enzymatic steps in that pathway are predicted to occur in many sequenced genomes, which MetaCyc does not [33]. MetaCyc labels individual pathways with information regarding the species in which their presence has been experimentally determined.

7. MetaCyc contains information about enzyme properties for specific species, like subunit composition, substrate specificity, cofactor requirements, activators and inhibitors. KEGG contains none of these information [33].

6.4.1 Biological aspects

Since MetaCyc is the union of individual species PGDBs, information about individual species is clearly separable. Viewing and analyzing pathways of several different species is supported by the pathway tools software. KEGG is a generic pathway database with merged information about many species. There is no clear separation of pathways based on species, but one can highlight parts of the pathways that occur in a given species. MetaCyc contains more comprehensive genomic information such as transcriptional units, regulatory proteins, transporters, promoters, and DNA binding sites in individual species, in addition to metabolic pathways. KEGG mainly consists of pathway information. MetaCyc contains several pathway variants and their maps for most pathways. KEGG pathway maps are manually constructed and include all connections in a pathway without clear separation of pathway variants. From Table 2.2, it is clear that KEGG is significantly ahead of MetaCyc in terms of the amount of substrate, reaction, and genetic information it offers. While MetaCyc has increased the number of species included, the data should be interpreted with caution as these are computationally derived pathway data.

6.4.2 File Organization

The flat files of the KEGG database are organized based on pathways, i.e., all the information needed to build a particular generic metabolic pathway are organized into five different file types and encompassed in a single directory. In Ecocyc, the data are organized according to species. All the pathways belonging to a particular species are stored in a single directory. Hence, there is data redundancy.

6.4.3 Database Schema

The main difference lies in the fact that KEGG can be readily reverse engineered into a relational database while reverse engineering MetaCyc into relational database is more complex as it is based on an object oriented data model. MetaCyc also has a Knowledge base system supporting the data stored within a Frame Knowledge Representation System (FRS) by validating the biological meaning of the stored data [30]. FRSs organize information within classes, which are collections of objects that share similar properties and attributes. Karp [30] proposed a schema for Ecocyc based on the class hierarchy shown in Figure 2.3. Both KEGG and Ecocyc are available in flat file formats. These flat files are generated from more structured databases that are not publicly accessible. We reverse engineer the two databases from their flat files to obtain all the tables, relations, and attributes in relational schema. We then select only those entities from the schema that are relevant in our context.

6.4.4 Web access features

Features provided by EcoCyc and MetaCyc on the web are listed below:

- In EcoCyc, each pathway map can be viewed in one of two levels of detail. The popular overview plus detail mantra of information visualization is utilized in this design.
- Much information is hidden behind the hyperlinks and not all details are visible even in the more detail view.
- Full names and not ID numbers are used on display. Hence, a map can appear cluttered when the names are often long. These names connect via hyperlinks to further details.
- Individual pathways link the compounds well but links between pathways are not viewable.

- User can view the structural and chemical composition of most compounds.
- Reactions with structures are printer friendly with larger pathways printable on multiple pages length wise.
- The distinction between links and non-links is not very clear due to the use of cleared fonts on the pathway maps.

Features provided by KEGG on the web are listed below:

- The KEGG website gives a very good "detailed overview" of entire pathway.
- The KEGG website provides good interconnection between different pathways.
- From the displayed pathway diagrams, not all names are obvious. It requires the user to click on the displayed entity ID numbers to even get to their names. This however leaves the pathway maps clear and easy to read.
- A clear distinction exists between different pathways and compounds present within any given pathway.
- Some KEGG pathways require horizontal scrolling and printing such pathways to a single letter sized paper is an issue.
- Web query options allow the users to retrieve all pathways and genes affected by a given (list of) enzyme(s). All completely sequenced genomes and some partially sequenced genomes are supported by the system.

Summary of the contents of EcoCyc model organism database is shown in Table 6.4. MetaCyc consists of metabolic pathways and enzymes from over 240 species. A summary of the MetaCyc contents as of December 1, 2004 is given in Table 6.5. KEGG consists of metabolic pathways and enzymes from over 174 species. A summary of the KEGG contents as of December 1, 2004 is given in Table 6.6. The KEGG database clearly outnumbers MetaCyc in terms of the number of compounds, reactions, genes, and pathways.

Object Class	Count
Pathways	182
Reactions	3676
Enzymes	1144
Transporters	197
Genes	4476
Transcription Units	977
Promoters	740
DNA Binding Sites	854
Citations	3508
DNA-Binding Transcriptional Regulators	100

Table 6.4: Summary of contents of EcoCyc database.

Object Class	Count
Metabolic Pathways	496
Reactions	4873
Enzymes	1665
Compounds	3465
Species	240
Citations	2381

Table 6.5: Summary of contents of MetaCyc database.

Object Class	Count	database name
pathways	10,677	PATHWAY
reference pathways	226	PATHWAY
ortholog tables	84	PATHWAY
organisms	174	GENOME
genes	481,325	GENES
KO assignments	3,415	KO
KO candidates	20,474	SSDB
chemical compounds	11,323	COMPOUND
chemical reactions	5,902	REACTION

Table 6.6: Summary of contents of KEGG database.

6.5 Glycolysis: a case study

Glycolysis is the sequence of reactions that convert glucose into pyruvate with the concomitant production of a relatively small amount of ATP. Gluconeogenesis is the biosynthesis of new glucose, i.e. not glucose from glycogen. The glycolysis pathway as it appears in KEGG is shown in Figure 6.2. In order to compare KEGG and MetaCyc, we compare glycolysis at the diagrammatic level and at the database level. At the diagrammatic level, in KEGG, glycolysis and gluconeogenesis are shown in the same pathway diagram but have separate database entries. There is a single database entry for the glycolysis pathway in KEGG.

The Glycolysis pathway as it appears in MetaCyc is shown in Figure at the URL [64]. Here, glycolysis and gluconeogenesis are separated both in pathway diagrams and in the database. There are three separate pathway diagrams for glycolysis namely, glycolysis, glycolysis2 and glycolysis3. The differences are in the starting substrate (glucose 6-phosphate versus beta-D-glucose or any other monosaccharide) and the ending substrates (acetate and lactate via pyruvate, versus L-alanine and acetate) distinguishing the positive and negative feedback effects within the pathway. There was no corresponding database entry for glycolysis3. Glycolysis pathway showing common elements in both KEGG and MetaCyc on the KEGG map is shown in the Figure 6.3. This analysis shows that EC numbers can be directly used in the process of matching up reactions and enzymes of KEGG and MetaCyc which in turn can be used in matching their substrates and pathways. Refer to Chapter 7.

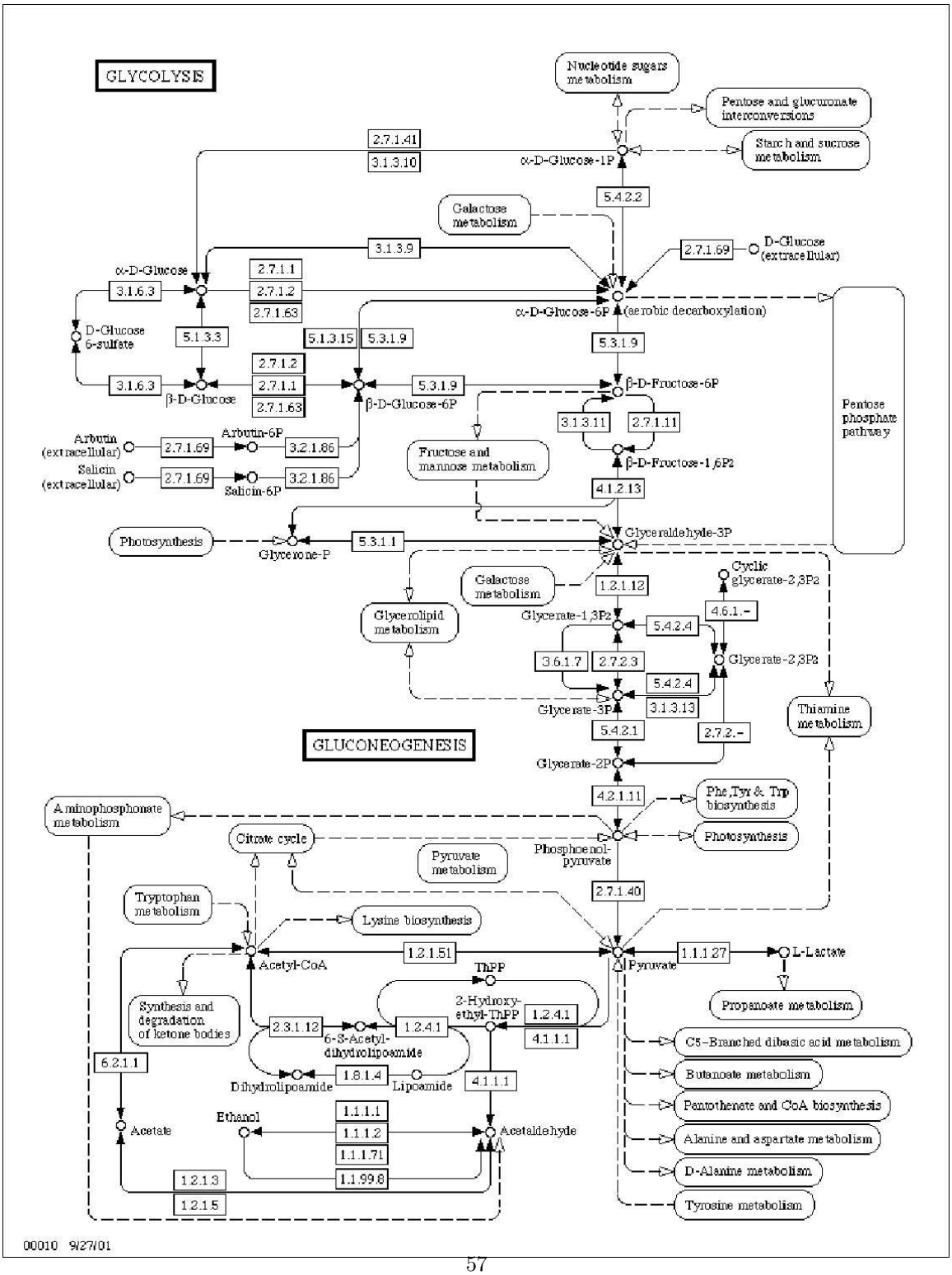

Figure 6.2: Glycolysis pathway in KEGG.

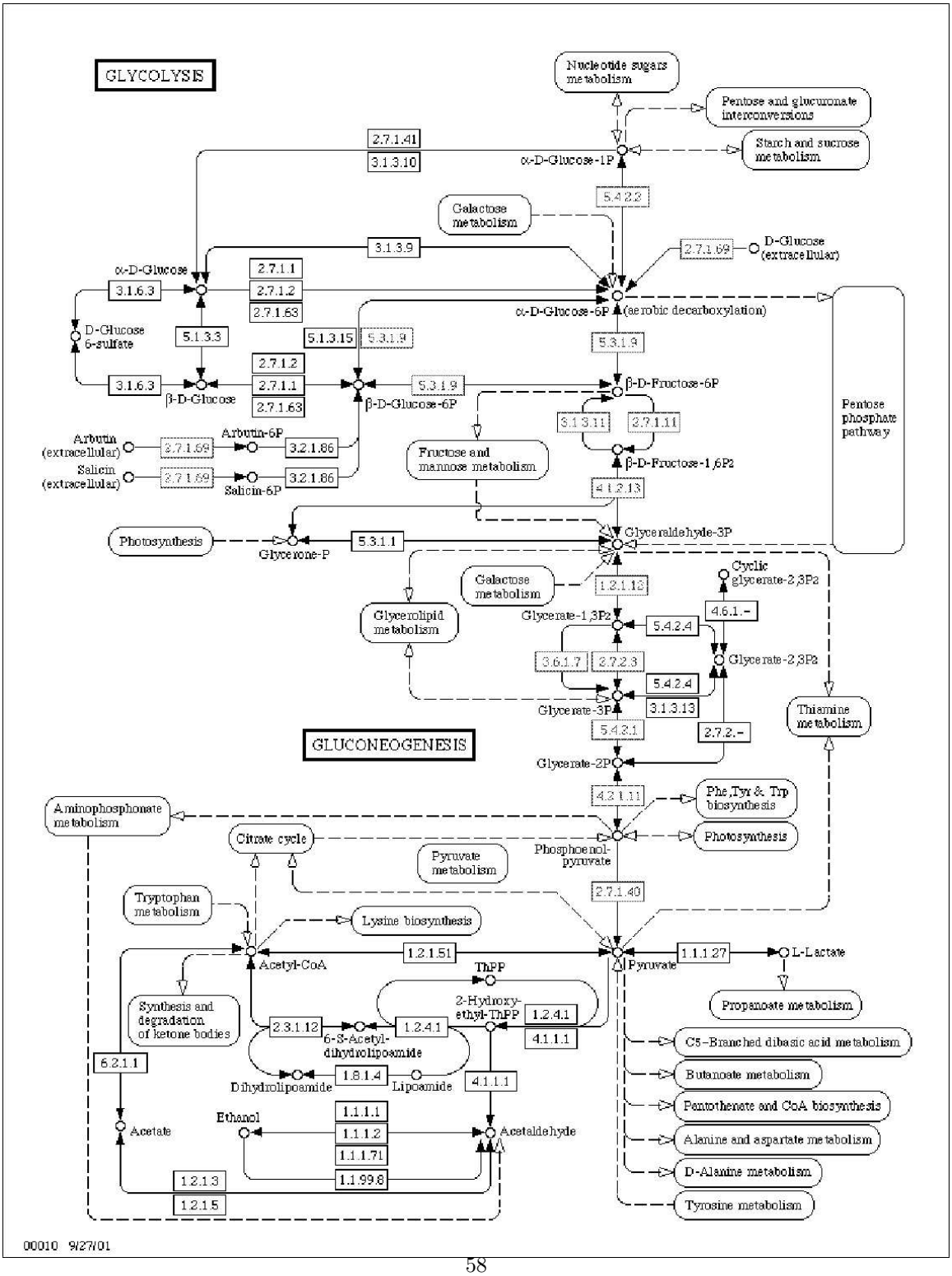

Figure 6.3: Glycolysis pathway in KEGG showing matches with that in MetaCyc in red.

Chapter 7

Conclusions and Future Direction

7.1 Conclusions

The creators of the pathway databases should thrust more emphasis on curating the databases with more consistent naming conventions. The database users should be aware of the various possible errors within these databases and the large inconsistencies between databases. Standardization is called for in naming various entities such as substrates, genes, and reactions. A standard classification scheme for biological pathways would be a significant help for users to obtain information about a particular pathway from various data sources. Identifying and fixing errors and inconsistencies within and between these databases requires domain experts (biological/biochemical scientists). This could prove costly in terms of time and effort. With WIT and EMP databases not being available lately for web access (for several months now), integrating BRENDA and ExPASy-biochemical pathways with our integrated system would mean current knowledge at your disposal.

7.2 Future Scope

This research is carried out as a part of an ITR grant [51] and lays the foundation for further research. The project aims to develop explanatory and predictive models of phenomena occurring within plant cells in response to drought and other abiotic stresses. By utilizing biological information from multiple sources such as experimental data (especially gene expression data); sequence, protein,

and other databases; and the biological literature, a new generation of biological models, called multimodal networks, will be developed. These models will represent multiple aspects of our current state of biological knowledge and of biological systems themselves such as, responses over time; response variation by subcellular compartments; uncertainty (our lack of complete knowledge of cell state); evolution of the genome; and the dynamic changes in biological information due to the boom in biological data and knowledge. With the emergence of XML based data exchange format standards for biochemical pathway data, such as, SBML, BioPAX, and KGML, future unification can be accomplished without having to create a local version of the component databases.

We propose the following improvements and enhancements to the PathMeld system:

1. Integrate BRENDA, WIT or ERGO, and ExPASy with PathMeld.
2. Support input and output of the PathMeld data in XML based emerging pathway exchange formats, namely, SBML, BioPAX, and KGML.
3. Addition of sub-cellular location and tissue specificity information will enable more accurate and detailed biological analysis.
4. Provide a powerful and biologist friendly web interface for the PathMeld system.
5. Maintain and update PathMeld whenever the component databases are upgraded.
6. Integration of PathMeld with standard genome databases such as GenBank will further help the biologists in the complete analysis of pathways.
7. Integrate the metabolic pathway information with signal transduction in plants.

7.3 Further challenges

PathMeld is a part of a much bigger initiative. The PathMeld system will be utilized in the ITR grant project for achieving the following tasks:

- To construct pathway diagrams based on a mathematical graph theoretic model of multimodal networks
- Develop novel algorithms to dynamically change and manipulate these pathways

- Perform automated layout of large pathways that exist in eukaryotic organisms
- Support methods for user navigation through large pathway networks
- Define standard ontologies for exchange of pathway data among different databases and application programs
- Create new analysis algorithms for extracting new insights from pathway networks. E.g., to aid drug design by analyzing diseased human pathway networks [26].

REFERENCES

[1] A Bairoch, *The ENZYME database in 2000*, Nucleic Acids Research, 28 (2000), pp. 304–305.

[2] T. Ahn, Y. Kim, S. Kim, D. Park, and H. Nam, *Building Integrated Pathway Genome Database in Cell Type Specific Manner - Encyclopedia of Human Liver*, Recomb, 46 (2000), p. 46.

[3] R. Apweiler, T. Attwood, A. Bairoch, A. Bateman, E. Birney, M. Biswas, P. Bucher, L. Cerutti, F. Corpet, M. Croning, R. Durbin, L. Falquet, W. Fleischmann, J. Gouzy, H. Hermjakob, N. Hulo, I. Jonassen, D. Kahn, A. Kanapin, Y. Karavidopoulou, R. Lopez, B. Marx, N. Mulder, T. Oinn, M. Pagni, F. Servant, C. Sigrist, and E. Zdobnov, *InterPro - an integrated documentation resource for protein families, domains, and functional sites*, Bioinformatics, 16 (2000), pp. 1145–1150.

[4] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro1, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L. L. Yeh, *UniProt: the Universal Protein knowledgebase*, Nucleic Acids Research, Database issue, 32 (2004), pp. D115–D119.

[5] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. S. holme, C. Yeats, and S. R. Eddy, *The Pfam Protein Families Database*, Nucleic Acids Research, 32 (2004), pp. D138–D141.

[6] A. D. Baxevanis, *The molecular biology database collection: 2003 update*, Nucleic Acids Research, 31 (2003).

[7] J. Bernhardt, K. Buttner, and J. Coppee, *The contribution of EC consortium to the two-dimensional protein index of Bacillus subtilis*, In Functional Analysis of Bacterial Genes: A Practical Manual, 1 (2001), pp. 63–74.

[8] J. Bernhardt, K. Buttner, C. Scharf, and M. Hecker, *Dual channel imaging of two-dimensional electropherograms in Bacillus subtilis*, Electrophoresis, 20 (1999), pp. 2225–2240.

[9] V. Biaudet, F. Samson, and P. Bessieres, *Micado: A network oriented database for microbial genomes*, Computational Applied Biosciences, 13 (1997), pp. 431–438.

[10] Biomolecular Relations in Information Transmission and Expression (BRITE) URL, *http://www.genome.ad.jp/brite/brite.html*.

[11] A. Bleasby, D. Akrigg, and T. Attwood, *OWL - A non-redundant, composite protein sequence database*, Nucleic Acids Research, 22(17) (1994), pp. 3574–3577.

[12] A. Brazma, P. Hingamp, and J. Quackenbush, *Minimum information about microarray exeriment (MIAME): towards standards for microarray data*, Nature Genetics, 29 (2001), pp. 365–371.

[13] A. Cuellar, C. Lloyd, P. Nielsen, D. Bullivant, D. Nikerson, and P. Hunter, *An Overview of CellML 1.1, a Biological Model Description Language*, SIMULATION, 79(12) (2003), pp. 740–747.

[14] N. Dutil, *Summary II- The Database KEGG and Scoring Biochemical Pathways*, (2002).

[15] Enzymes and Metabolic Pathways (EMP) URL, *http://www.empproject.com/*.

[16] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. Sigrist, K. Hofmann, and A. Bairoch, *The PROSITE database, its status in 2002*, Nucleic Acids Research, 30 (2002), pp. 235–238.

[17] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R. Appel, and A. Bairoch, *ExPASy: the proteomics server for in-depth protein knowledge and analysis*, Nucleic Acids Research, 31 (2003), pp. 3784–3788.

[18] B. GD, B. D, and H. CW, *BIND: the Biomolecular Interaction Network Database*, Nucleic Acids Research, 31(1) (2003), pp. 248–250.

[19] C. R. Harwood and I. Moszer, *From gene reguation to gene funnction: regulatory networks in Bacillus subtilis*, Comparative and Functional Genomics, 3 (2002), pp. 37–41.

[20] M. Hoebeke, H. Chiapello, P. Noirot, and P. Bessieres, *SPID: A Bacillus subtilis protein interaction database*, Bioinformatics, 17(12) (2001), pp. 1209–1212.

[21] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. L. Novere, L. M. Loew, D. Lucio, P. Mendes, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang, *The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models*, Bioinformatics, 19(4) (2003), pp. 524–531.

[22] X. I, R. DW, S. L, B. MK, M. EM, and E. D, *DIP: The Database of Interacting Proteins*, Nucleic Acids Research, 28 (2000), pp. 289–291.

[23] T. Ideker, T. Galitski1, and L. Hood, *A new approach to decoding life: Systems Biology*, Annual Review of Genomics and Human Genetics, 02 (2003), pp. 343–372.

[24] M. Kanehisa, *Databases of biological information*, Trends Guide to Bioinformatics, (1998), pp. 24–26.

[25] M. Kanehisa and S. Goto, *KEGG: Kyoto Encyclopedia of Genes and Genomes*, Nucleic Acids Research, 28 (2000), pp. 27–30.

[26] P. D. Karp, *Pathway Databases: A Case Study in Computational Symbolic Theories*, Science, 293 (2001), pp. 2040–2044.

[27] P. D. Karp, M. Krummenacker, S. Paley, and J. Wagg, *Integrated pathway/genome databases and their role in drug discovery*, Trends in Biotechnology, 17(7) (1999), pp. 275–281.

[28] P. D. Karp and S. Paley, *Integrated Access to Metabolic and Genomic Data*, Computational Biology, (1996).

[29] P. D. Karp, S. Paley, and P. Romero, *The pathway tools software*, Bioinformatics, 18 (2002), pp. S1–S8.

[30] P. D. Karp and M. Riley, *Ecocyc: The resource and the lessons learned*, Bioinformatics, (1999).

[31] P. D. Karp, M. Riley, S. M. Paley, and A. P. Toole, *The MetaCyc Database*, Nucleic Acids Research, 30 (2002), pp. 59–61.

[32] P. D. Karp, M. Riley, M. Saier, I. Paulsen, S. Paley, and A. Pellegrini-Toole, *The EcoCyc and MetaCyc databases*, Nucleic Acids Research, 28 (2000), pp. 56–59.

[33] P. D. Karp, M. Riley, M. Saier, I. T. Paulsen, J. Collado-Vides, S. M.Paley, A. Pellegrini-Toole, C. Bonavides, and S. Gama-Castro, *The EcoCyc Database*, Nucleic Acids Research, 30 (2002), pp. 56–61.

[34] R. Kuffner, R. Zimmer, and T. Lengauer, *Pathway analysis in metabolic databases via differential metabolic display (DMD)*, Bioinformatics, 16 (2000), pp. 825–836.

[35] Z. Lacroix, *Biological Data Integration: Wrapping Data and Tools*, IEEE Transactions on Information Technology in Biomedicine, 6 (2002), pp. 123–128.

[36] W. Li and C. Clifton, *SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks*, Data and Knowledge Engineering, 33 (2000), pp. 49–84.

[37] E. Lim and R. H. Chiang, *The integration of relationship instances from heterogeneous databases*, Decision Support Systems, 29 (2000), pp. 153–167.

[38] G. Lyon, A. Newton, and B. Marshall, *The need for a standard nomenclature for gene classification and a generic, automated tool to assist in hypothesis formulation in cell signalling*, Plant Pathology, 3(2) (2002), pp. 103–109.

[39] J. Macauley, H. Wang, and N. Goodman, *A model system for studying the integration of molecular biology databases*, Bioinformatics, 14 (1998), pp. 575–582.

[40] C. D. Maranas and A. P. Burgard, *Web site review: Review of EcoCyc and MetaCyc Databases*, Metabolic Engineering, 3 (2001), pp. 98–99.

[41] N. Matoba, J. Tanoue, M. Yoshikawa, and S. Uemura, *A System for Integration of Heterogeneous Biological XML Data*, Genome Informatics, 12 (2001), p. 2.

[42] I. Moszer, L. Jones, S. Moreira, C. Fabry, and A. Danchin, *SubtiList: The reference database for the Bacillus subtilis genome*, Nucleic Acids Research, 30(1) (2002), pp. 62–65.

[43] M. Nakao, H. Bono, S. Kawashima, T. Kamiya, K. Sato, S. Goto, and M. Kanehisa, *Genome-scale Gene Expression Analysis and Pathway Reconstruction in KEGG*.

[44] R. Overbeek, N. Larson, G. D. Pusch, M. D'Souza, E. S. Jr, N. Kyrpides, M. Fonstein, N. Maltsev, and E. Selkov, *WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction*, Nucleic Acids Research, 28(1) (2000), pp. 123–125.

[45] S. M. Paley and P. D. Karp, *Adapting EcoCyc for use on the World Wide Web*, Gene, 172 (1996), pp. GC43–GC50.

[46] PangeaSystems, *Pathway Tools User's Guide Version 7.0*, Copyright SRI international, 1 (1996), p. 100.

[47] B. Parvin, Q. Yang, G. Fontenay, M. Helen, and B. Hoff, *BioSig: An Imaging Bioinformatic System for Studying Phenomics*, IEEE, 02 (2002), pp. 65–72.

[48] S. Paxia, A. Rudra, Y. Zhou, and B. Mishra, *A random walk down the genomes: DNA Evolution in Valis*, IEEE, 2 (2002), pp. 73–79.

[49] C. Petri, *Fundamentals of a Theory of Asynchronous Information Flow*, Proc. of IFIP Congress, 62 (1963), pp. 386–390.

[50] PTOOLS software environment for accessing, querying and manipulating metacyc database URL, *http://bioinformatics.ai.sri.com/ptools/*.

[51] N. Ramakrishnan, R. G. Alscher, L. S. Heath, L. Watson, and J. Weller, *Understanding Stress Resistance Mechanisms in Plants: Multimodal Models Integrating Experimental Data, Databases, and the Literature*, (2001).

[52] F. Samson, V. Biaudet-Brunaud, and S. Gas, *Micado: An integrative database dedicated to the functional analysis of Bacillus subtilis and microbial genomics*, In Functional Analysis of Bacterial Genes: A Practical Manual, 1 (2001), pp. 45–52.

[53] M. A. S. Saqi and M. J. E. Sternberg, *A Structural Census of Metabolic Networks for E. coli*, Journal of Molecular Biology, 313 (2001), pp. 1195–1206.

[54] K. Sato, T. Katsurada, Y. Kimura, and M. Kanehisa, *Integrated GENES Database in KEGG*.

[55] I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, and D. Schomburg, *BRENDA, the enzyme database: updates and major new developments*, Nucleic Acids Research, 32 (2004), pp. D431–D433.

[56] Signaling Pathway Database (SPAD) URL, *http://www.grt.kyushu-u.ac.jp/spad/*.

[57] W. Sujansky, *Heterogeneous Database Integration in Biomedicine*, Journal of Biomedical Informatics, 34 (2001), pp. 285–298.

[58] The BioCyc database collection URL, *http://www.biocyc.org*.

[59] The Biological Pathways Exchange (BioPax) URL, *http://www.biopax.org/index.html*.

[60] The Cell Signaling Networks Database (CSNDB) URL, *http://geo.nihs.go.jp/cgi-bin/model/wwwace*.

[61] The Database resource for analysis of signal transduction in cells (DRASTIC) URL, *http://www.drastic.org.uk/*.

[62] The EcoCyc Eschericia coli model organism specific database URL, *http://ecocyc.org/*.

[63] The International Union of Biochemistry and Molecular Biology (IUBMB) URL, *http://www.chem.qmul.ac.uk/iubmb/*.

[64] The URL for glycolysis pathway in MetaCyc database., *http://biocyc.org/META/new-image?type=PATHWAYobject=GLYCOLYSISdetail-level=2*.

[65] The URL for the BRENDA resource, *http://www.brenda.uni-koeln.de*.

[66] U. Wittig and A. D. Beuckelaer, *Analysis and comparison of metabolic pathway databases*, Briefings in Bioinformatics, 2 (2001), pp. 126–142.

[67] J. Wixon, *Pathway Databases*, Comparative and Functional Genomics, 2 (2001), pp. 391–397.

[68] Y. Xia, R. E. Stinner, and P. Chu, *Database integration with the web for biologists to share data and information*, Electronic Journal of Biotechnology, 5 (2002), pp. 154–161.

Appendix A

Sample source code for flat file parser

This and the next few appendices provide samples of the source code developed for the implementation of different components of the PathMeld methodology.

```
/*
 * filename:    get_Metapathway.c
 * author:      Harsha K. Rajasimha
 * description: This parses the Metacyc/pathways.dat flatfile from the
 *              Metacyc database and outputs a tab delimited file
 *              containing pathway-ID and Common name.
 *              The latter files will be used to populate the
 *              Metacyc database in postgres.
 */
# include <stdio.h>
# include <string.h>
# define BUFSZ 1024
# define LEN 1024

FILE *Metacyc_pathway;
char buffer[BUFSZ];

void trim(char *str) {
    printf(``in trim");
    char *tmpstr;
    str[strlen(str)-1] = '\0';
    tmpstr = strstr(str,`` - ");
```

```
    strcpy(str,(tmpstr)+3);
}

int parse(void) {
   char pwyid[LEN];
   char pname[LEN];
   pwyid[0] = '\0';
   pname[0] = '\0';
   fgets(buffer,LEN-1,stdin);

   while (strncmp(buffer,``//",2) != 0) {
           if (strncmp(``UNIQUE-ID",buffer,9) == 0) {
            trim(buffer);
            strcpy(pwyid,buffer);
            fgets(buffer,LEN-1,stdin);
       } else
       if (strncmp(``COMMON-NAME",buffer,11) == 0) {
            trim(buffer);
            strcpy(pname,buffer);
            fgets(buffer,LEN-1,stdin);
       }
       else
       if (buffer[0] != ' ') {
            /* gobble it */
            fgets(buffer,LEN-1,stdin);
            while (buffer[0] == ' ') {
                /* gobble it */
                fgets(buffer,LEN-1,stdin);
            }
       }
   }
   printf(``\n");
   fprintf(Metacyc_pathway,``%s\t%s\n", pwyid, pname);

   return 0;
}

int main(void) {
   int done = 0;
   printf(``in main");

   if ((Metacyc_pathway = fopen(``Metacyc_pathway.sql",``wb")) == NULL) {
       fprintf(stderr,``(error) i can't open pathways.dat\n");
       return 1;
   }
   printf(``in Main File Opened");
```

```
    while (!done) {
        done = parse();
        if (feof(stdin)) break;
    }
    fclose(Metacyc_pathway);

    return 0;
}
```

Note that this source file serves as a template and with little changes, this code can be utilized in parsing all MetaCyc flat files.

Appendix B

SQL code for PathMeld database creation

```
$ psql

$ psql create database pathmeld;

$ psql \q

$ psql pathmeld < kegg_schema.sql

$ psql pathmeld < metacyc_schema.sql

The Makefile in the directory in which the source files exist executes
all programs that parse relevant flat files to generate the sql files
with tab delimited data for each of the tables in the unified schema.
Each of these sql files are appended with an insert command at the
beginning:

$ psql pathmeld < kegg_table.sql

and a terminate command at the end of file:

\.

A script file executes all these sql commands for populating the data
in the thus generated sql files into the unified database tables.
```

(Thanks to Allan Sioson for providing me with the KEGG tables).

```
-- PostgreSQL database dump

\connect - hrajasim
```

```
-- Name: MetaCyc; Type: SCHEMA; Schema: -;

CREATE SCHEMA MetaCyc;

-- Name: unify; Type: SCHEMA; Schema: -;

CREATE SCHEMA unify;

-- Name: KEGG; Type: SCHEMA; Schema: -;

CREATE SCHEMA KEGG;

SET search_path = MetaCyc, pg_catalog;

-- Name: class; Type: TABLE; Schema: MetaCyc;

CREATE TABLE ‘‘class" (
    clsid character varying(254) NOT NULL,
    ‘‘type" character varying(254),
    cname character varying(254)
);

-- Name: compound; Type: TABLE; Schema: MetaCyc;

CREATE TABLE compound (
    compoundid character varying(254) NOT NULL,
    cname character varying(254),
    chemicalformula character varying(254),
    newformulaformat character varying(254)
);

-- Name: cpdnames; Type: TABLE; Schema: MetaCyc;

CREATE TABLE cpdnames (
    compoundid character varying(254),
    synonym character varying(254)
);

-- Name: protein; Type: TABLE; Schema: MetaCyc;

CREATE TABLE protein (
    proteinid character varying(254) NOT NULL,
    cname character varying(254),
    ‘‘type" character varying(254)
);
```

```
-- Name: reaction; Type: TABLE; Schema: MetaCyc;

CREATE TABLE reaction (
    reactionid character varying(254) NOT NULL,
    cname character varying(254),
    ‘‘type" character varying(254),
    ecnumber character varying(254)
);

-- Name: enzrxn; Type: TABLE; Schema: MetaCyc;

CREATE TABLE enzrxn (
    enzrxnid character varying(254) NOT NULL,
    enzyme character varying(254),
    rdirection character varying(254),
    reaction character varying(254)
);

-- Name: enzrxn_enzyme; Type: TABLE; Schema: MetaCyc;

CREATE TABLE enzrxn_enzyme (
    enzrxnid character varying(254),
    enzyme character varying(254)
);

-- Name: reaction_eqn; Type: TABLE; Schema: MetaCyc;

CREATE TABLE reaction_eqn (
    rxnid character varying(254),
    rxnleft character varying(254),
    rxnright character varying(254)
);

SET search_path = public, pg_catalog;

-- Name: gene; Type: TABLE; Schema: public;

CREATE TABLE gene (
    geneid character varying(254) NOT NULL,
    genetype character varying(254),
    cname character varying(254),
    product character varying(254),
    producttype character varying(254)
);
```

```
SET search_path = MetaCyc, pg_catalog;

-- Name: gene; Type: TABLE; Schema: MetaCyc;

CREATE TABLE gene (
    geneid character varying(254) NOT NULL,
    genetype character varying(254),
    cname character varying(254)
);

-- Name: activator; Type: TABLE; Schema: MetaCyc;

CREATE TABLE activator (
    erxnid character varying(254),
    activatorid character varying(254),
    activatortype character varying(254)
);

-- Name: inhibitor; Type: TABLE; Schema: MetaCyc;

CREATE TABLE inhibitor (
    erxnid character varying(254),
    inhibitorid character varying(254),
    inhibitortype character varying(254)
);

-- Name: proteinnames; Type: TABLE; Schema: MetaCyc;

CREATE TABLE proteinnames (
    protid character varying(254),
    synonym character varying(254)
);

-- Name: gene_product; Type: TABLE; Schema: MetaCyc;

CREATE TABLE gene_product (
    genid character varying(254),
    prodid character varying(254),
    prodtype character varying(254)
);

-- Name: rxn_enzrxn; Type: TABLE; Schema: MetaCyc;

CREATE TABLE rxn_enzrxn (
    rxnid character varying(254),
    erxnid character varying(254)
```

```
);

-- Name: reaction_in_pwy; Type: TABLE; Schema: MetaCyc;

CREATE TABLE reaction_in_pwy (
    rxnid character varying(254),
    pwyid character varying(254)
);

-- Name: protein_enzyme; Type: TABLE; Schema: MetaCyc;

CREATE TABLE protein_enzyme (
    protid character varying(254),
    catalyzes character varying(254)
);

SET search_path = unify, pg_catalog;

-- Name: compound; Type: TABLE; Schema: unify;

CREATE TABLE compound (
    keggid character varying(254),
    metacycid character varying(254),
    kname character varying(254),
    mname character varying(254),
    knamecount integer,
    matchcount integer,
    mnamecount integer
);

SET search_path = KEGG, pg_catalog;

-- Name: compound; Type: TABLE; Schema: KEGG;

CREATE TABLE compound (
    entry character varying(255) NOT NULL,
    name text,
    formula character varying(255)
);

-- Name: compound_alias; Type: TABLE; Schema: KEGG;

CREATE TABLE compound_alias (
    entry character varying(255),
    name text
);
```

```
-- Name: enzyme; Type: TABLE; Schema: KEGG;

CREATE TABLE enzyme (
    entry character varying(255) NOT NULL,
    obsolete character(1),
    name text,
    ‘‘class" character varying(255),
    class_desc character varying(255),
    sysname text
);

-- Name: enzyme_alias; Type: TABLE; Schema: KEGG;

CREATE TABLE enzyme_alias (
    entry character varying(255),
    name text
);

-- Name: genes_encode_enzyme; Type: TABLE; Schema: KEGG;

CREATE TABLE genes_encode_enzyme (
    e_entry character varying(255),
    o_entry character varying(255),
    genecode character varying(255),
    genename character varying(255)
);

-- Name: reaction; Type: TABLE; Schema: KEGG;

CREATE TABLE reaction (
    entry character varying(255) NOT NULL,
    name text,
    definition text,
    equation character varying(255)
);

-- Name: r_needs_e; Type: TABLE; Schema: KEGG;

CREATE TABLE r_needs_e (
    r_entry character varying(255),
    e_entry character varying(255)
);

-- Name: pathway; Type: TABLE; Schema: KEGG;
```

```
CREATE TABLE pathway (
    entry character varying(255) NOT NULL,
    name character varying(255)
);

-- Name: p_has_r; Type: TABLE; Schema: KEGG;

CREATE TABLE p_has_r (
    p_entry character varying(255),
    r_entry character varying(255)
);

-- Name: edge; Type: TABLE; Schema: KEGG;

CREATE TABLE edge (
    r_entry character varying(255),
    s1_c_entry character varying(255),
    s2_c_entry character varying(255),
    direction character varying(255)
);

-- Name: edge2; Type: TABLE; Schema: KEGG;

CREATE TABLE edge2 (
    r_entry character varying(255),
    s1_c_entry character varying(255),
    s2_c_entry character varying(255),
    direction character varying(255)
);

-- Name: organism; Type: TABLE; Schema: KEGG;

CREATE TABLE organism (
    entry character varying(255) NOT NULL,
    name character varying(255)
);

SET search_path = MetaCyc, pg_catalog;

-- Name: cpd_unify; Type: TABLE; Schema: MetaCyc;

CREATE TABLE cpd_unify (
    entry character varying(255),
    compoundid character varying(254),
    name text
);
```

```
-- Name: cpd_unify2; Type: TABLE; Schema: MetaCyc;

CREATE TABLE cpd_unify2 (
    entry character varying(255),
    compoundid character varying(254),
    name text
);

-- Name: cpd_unify1; Type: TABLE; Schema: MetaCyc;

CREATE TABLE cpd_unify1 (
    entry character varying(255),
    compoundid character varying(254),
    name text
);

-- Name: cpd_unified; Type: TABLE; Schema: MetaCyc;

CREATE TABLE cpd_unified (
    entry character varying(255),
    compoundid character varying(254),
    name text
);

-- Name: ninetykegg; Type: TABLE; Schema: MetaCyc;

CREATE TABLE ninetykegg (
    entry character varying(255),
    name text
);

-- Name: sixtyfivemetacyc; Type: TABLE; Schema: MetaCyc;

CREATE TABLE sixtyfivemetacyc (
    compoundid character varying(254),
    synonym character varying(254)
);

-- Name: cpd_unified_by_frm; Type: TABLE; Schema: MetaCyc;

CREATE TABLE cpd_unified_by_frm (
    entry character varying(255),
    compoundid character varying(254),
    name text
);
```

```
-- Name: additional_recs; Type: TABLE; Schema: MetaCyc;

CREATE TABLE additional_recs (
    entry character varying(255),
    compoundid character varying(254),
    name text
);

-- Name: cpd_unified_plus; Type: TABLE; Schema: MetaCyc;

CREATE TABLE cpd_unified_plus (
    entry character varying(255),
    compoundid character varying(254),
    name text
);

-- Name: mrequired; Type: TABLE; Schema: MetaCyc;

CREATE TABLE mrequired (
    compoundid character varying(254),
    newformulaformat character varying(254)
);

-- Name: krequired; Type: TABLE; Schema: MetaCyc;

CREATE TABLE krequired (
    entry character varying(255),
    name text,
    formula character varying(255)
);

-- Name: compinpwy; Type: TABLE; Schema: MetaCyc;

CREATE TABLE compinpwy (
    compoundid character varying(254),
    synonym character varying(254),
    pathwayid character varying(254)
);

-- Name: noformulakm; Type: TABLE; Schema: MetaCyc;

CREATE TABLE noformulakm (
    entry character varying(255),
    name text,
    compoundid character varying(254)
```

```
);

-- Name: pathway; Type: TABLE; Schema: MetaCyc;

CREATE TABLE pathway (
    pid character varying(254),
    ptype character varying(254),
    pname character varying(254)
);

-- Name: pathway_rlist; Type: TABLE; Schema: MetaCyc;

CREATE TABLE pathway_rlist (
    pid character varying(254),
    rlist character varying(254)
);

-- Name: pathway_synonym; Type: TABLE; Schema: MetaCyc;

CREATE TABLE pathway_synonym (
    pid character varying(254),
    ‘‘names" character varying(254)
);

-- Name: pathway_inpwy; Type: TABLE; Schema: MetaCyc;

CREATE TABLE pathway_inpwy (
    pid character varying(254),
    inpwy character varying(254)
);

-- Name: reaction_left; Type: TABLE; Schema: MetaCyc;

CREATE TABLE reaction_left (
    rid character varying(254),
    lcompounds character varying(254)
);

-- Name: reaction_right; Type: TABLE; Schema: MetaCyc;

CREATE TABLE reaction_right (
    rid character varying(254),
    rcompounds character varying(254)
);

SET search_path = public, pg_catalog;
```

```
-- Name: genes_in_KEGG_pathways; Type: TABLE; Schema: public;

CREATE TABLE genes_in_KEGG_pathways (
    entry character varying(255),
    name character varying(255),
    o_entry character varying(255),
    genecode character varying(255),
    genename character varying(255)
);

-- Name: genes_in_MetaCyc_pathways; Type: TABLE; Schema: public;

CREATE TABLE genes_in_MetaCyc_pathways (
    pid character varying(254),
    ptype character varying(254),
    pname character varying(254),
    o_entry character varying(255),
    genecode character varying(255),
    genename character varying(255)
);

-- Name: KEGG_pathway_links; Type: TABLE; Schema: public;

CREATE TABLE KEGG_pathway_links (
    pid_a character varying(255),
    pid_b character varying(255),
    species character varying(255),
    genecode character varying(255)
);

-- Name: MetaCyc_pathway_links; Type: TABLE; Schema: public;

CREATE TABLE MetaCyc_pathway_links (
    pid_a character varying(254),
    pid_b character varying(254),
    species character varying(255),
    genecode character varying(255)
);

SET search_path = MetaCyc, pg_catalog;

-- Name: class_pkey; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY ‘‘class"
    ADD CONSTRAINT class_pkey PRIMARY KEY (clsid);
```

```
-- Name: compound_pkey; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY compound
    ADD CONSTRAINT compound_pkey PRIMARY KEY (compoundid);

-- Name: $1; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY cpdnames
    ADD CONSTRAINT ``$1" FOREIGN KEY (compoundid) REFERENCES compound(compoundid)
           ON UPDATE NO ACTION ON DELETE NO ACTION;

-- Name: protein_pkey; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY protein
    ADD CONSTRAINT protein_pkey PRIMARY KEY (proteinid);

-- Name: reaction_pkey; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY reaction
    ADD CONSTRAINT reaction_pkey PRIMARY KEY (reactionid);

-- Name: enzrxn_pkey; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY enzrxn
    ADD CONSTRAINT enzrxn_pkey PRIMARY KEY (enzrxnid);

-- Name: $1; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY enzrxn
    ADD CONSTRAINT ``$1" FOREIGN KEY (enzyme) REFERENCES protein(proteinid)
           ON UPDATE NO ACTION ON DELETE NO ACTION;

-- Name: $1; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY enzrxn_enzyme
    ADD CONSTRAINT ``$1" FOREIGN KEY (enzrxnid) REFERENCES enzrxn(enzrxnid)
           ON UPDATE NO ACTION ON DELETE NO ACTION;

-- Name: $2; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY enzrxn_enzyme
    ADD CONSTRAINT ``$2" FOREIGN KEY (enzyme) REFERENCES protein(proteinid)
           ON UPDATE NO ACTION ON DELETE NO ACTION;

-- Name: $1; Type: CONSTRAINT; Schema: MetaCyc;
```

```
ALTER TABLE ONLY reaction_eqn
    ADD CONSTRAINT ‘‘$1" FOREIGN KEY (rxnid) REFERENCES reaction(reactionid)
           ON UPDATE NO ACTION ON DELETE NO ACTION;

SET search_path = public, pg_catalog;

-- Name: gene_pkey; Type: CONSTRAINT; Schema: public;

ALTER TABLE ONLY gene
    ADD CONSTRAINT gene_pkey PRIMARY KEY (geneid);
SET search_path = MetaCyc, pg_catalog;

-- Name: gene_pkey; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY gene
    ADD CONSTRAINT gene_pkey PRIMARY KEY (geneid);
SET search_path = KEGG, pg_catalog;

-- Name: compound_pkey; Type: CONSTRAINT; Schema: KEGG;

ALTER TABLE ONLY compound
    ADD CONSTRAINT compound_pkey PRIMARY KEY (entry);

-- Name: enzyme_pkey; Type: CONSTRAINT; Schema: KEGG;

ALTER TABLE ONLY enzyme
    ADD CONSTRAINT enzyme_pkey PRIMARY KEY (entry);

-- Name: reaction_pkey; Type: CONSTRAINT; Schema: KEGG;

ALTER TABLE ONLY reaction
    ADD CONSTRAINT reaction_pkey PRIMARY KEY (entry);

-- Name: pathway_pkey; Type: CONSTRAINT; Schema: KEGG;

ALTER TABLE ONLY pathway
    ADD CONSTRAINT pathway_pkey PRIMARY KEY (entry);

SET search_path = MetaCyc, pg_catalog;

-- Name: $1; Type: CONSTRAINT; Schema: MetaCyc;

ALTER TABLE ONLY pathway_rlist
    ADD CONSTRAINT ‘‘$1" FOREIGN KEY (rlist) REFERENCES reaction(reactionid)
        ON UPDATE NO ACTION ON DELETE NO ACTION;
```

Appendix C

Database population

To each of the files generated by the parsing step, just add the following statement at the first line of the file:

```
copy Metacyc.pathway from stdin;
```

```
and terminate that file by ‘‘\."
```

Note that Metacyc is a schema within the database to separate its tables logically from KEGG tables.

Appendix D

Examples of approximate match from agrep

In this Chapter, we present the example output of substrate name matching from agrep. Note that we consider here only those MetaCyc substrates that remain unmatched after the exact name and chemical formula matching is applied. This is how the agrep output has to be interpreted: The string enclosed within the double quotes following the string "PATTERN: " is the name of a MetaCyc substrate that falls in the 17.4% of unmatched MetaCyc substrates after exact name and chemical formula matching is applied. The lines of strings that follow the "PATTERN: " line are the KEGG substrate names (all KEGG names considered here) that are guessed to be matches to the MetaCyc name in the pattern. Note that the MetaCyc substrates referred to by the MetaCyc names listed in Appendix D.3 are absent in the KEGG database.

D.1 Positive matching examples

```
PATTERN: "(1, 4-beta-d-xylan)n"
(1,4-beta-d-xylan)n
(1,4-beta-d-xylan)n+1

PATTERN: "(aminooxy)acetate"
gamma-aminooxaloacetate

PATTERN: "(aminooxy)acetic acid"
aminoacetic acid
benzoylaminoacetic acid

PATTERN: "(s)"
(s)
(s)-alanine
alpha,4(s),6-beta,16-beta)-
oxoprolyl]-(s)-alanine benzyl ester
(+)-(s)-carvone
```

```
(-)-(s)-limonene

PATTERN: "--abietadiene"
(-)-abietadiene

PATTERN: "1, 4-beta-d-xylan1"
(1,4-beta-d-xylan)n
(1,4-beta-d-xylan)n+1
1,4-beta-d-xylan

PATTERN: "1,2-diglyceride"
2-glyceride
diglyceride
monogalactosyldiglyceride
monoglucosyldiglyceride
```

D.2 Negative matching examples

```
PATTERN: "(3-methylbenzyl)succinyl-coa"
(hydroxymethylphenyl)succinyl-coa

PATTERN: "(6r)-6-fluoro-epsp"
(6r)-6-(l-erythro-1,2-dihydroxypropyl)-5,6,7,8-tetrahydro-4a-
(6r)-6-(l-erythro-1,2-dihydroxypropyl)-7,8-dihydro-6h-pterin

PATTERN: "(e)-2-nonen-1-al"
(e)-2-benzylidenesuccinyl-coa
(e)-2-butenyl-4-methyl-threonine
(e)-2-decene-4,6,8-triynoic acid methyl ester
(e)-2-methyl-4-(1h-purin-6-ylamino)but-2-en-1-ol
3-methyl-2-buten-1-ol
3-phenyl-2-propen-1-ol
5-d-(5/6)-5-c-(hydroxymethyl)-2,6-dihydroxycyclohex-2-en-1-one
prop-2-yne-1-ol

PATTERN: "(e)-2-nonenal"
(e)-2-butenyl-4-methyl-threonine

PATTERN: "1,2-ethanediamine"
1,4-butanediamine
1,5-pentanediamine
n,n'-bis(3-aminopropyl)-1,4-butanediamine

PATTERN: "1-aminoadipate-3-s-ald"
2-aminoadipate 6-semialdehyde
l-2-aminoadipate 6-semialdehyde
```

```
PATTERN: "1-aminoadipate-3-semialdehyde"
2-aminoadipate 6-semialdehyde
l-2-aminoadipate 6-semialdehyde

PATTERN: "1-ethyladenine"
1-methyladenine
trna containing n1-methyladenine
```

D.3 MetaCyc substrates that do not match any of the KEGG substrates

```
(-)threo-1-hydroxy-1,2,5-pentane-tricarboxylate
(-)threo-1-hydroxy-1,2,5-pentane-tricarboxylic acid
(-)threo-1-hydroxy-1,2,6-hexanetricarboxylate
(-)threo-1-hydroxy-1,2,6-hexanetricarboxylic acid
(-)threo-iso(homo)2citrate
(-)threo-iso(homo)3citrate
(e)-4-[[3-(acetylhydroxyamino)propyl]-amino]-2-hydroxy-2-[2-[3-[hydroxy(1-oxo-2-
        decenyl)amino]propyl]amino]-2-oxoethyl]-4-oxobutanoic acid
(24r)-24-methyl-cholest-4-en-3-one
(24r)-24-methyl-cholest-4-en-3beta-ol
(3-methylphenyl)itaconyl-coa
(6r)-6-fluoro-5-enolpyruvylshikimate 3-phosphate
(6s)-6-fluoro-5-enolpyruvylshikimate-3-phosphate
(6s)-6-fluoro-epsp
(r)-2-hydroxy-1,2,5-pentanetricarboxylate
(r)-2-hydroxy-1,2,5-pentanetricarboxylic acid
(r)-2-hydroxy-1,2,6-hexanetricarboxylate
(r)-2-hydroxy-1,2,6-hexanetricarboxylic acid
(r)-6-[l-2-(2-amino-1,4,5,6-tetrahydro-4-pyrimidinyl)glycine]viomycin
(z)-1,2,5-pent-1-enetricarboxylic acid
(z)-1,2,6-hex-1-enetricarboxylic acid
1,2,3-propanetriol 1,2-dinitrate
1,2,3-propanetriol 1,3-dinitrate
1,2-diacyl-3-o-(alpha-d-glucopyranosyl(1->2)-o-alpha-d-glucopyranosyl)sn-glycerol
1,2-dihydroxy-3-keto-5-methylthiopentane
1,2-dihydroxy-3-keto-5-methylthiopentene
1,2-dihydroxy-3-keto-5-methylthiopentene anion
1,3-biphosphoglyceric acid
1,3-diphosphoglyceric-acid
1-(4-amino-2-methylpyrimid-5-ylmethyl)-3-(beta-hydroxyethyl)-2-methylpyridinium
        bromide
1-(4-hydroxy-2-methylpyrimid-5-ylmethyl)-3-(beta-hydroxyethyl)-2-methylpyridinium
        bromide
```

1-(l-threo-3,6-diamino-4-hydroxyhexanoic acid)viomycin
1-18:1-2-16:0-monogalactosyldiacylglycerol
1-18:1-2-16:0-sn-glycerol-3-phosphate
1-18:1-2-16:1-monogalactosyldiacylglycerol
1-18:1-2-18:1-sn-glycerol-3-phosphocholine
1-18:1-2-trans-16:1-sn-glycerol-3-phosphate
1-18:2-2-16:2-monogalactosyldiacylglycerol
1-18:2-2-18:2-sn-glycerol-3-phosphocholine
1-18:2-2-trans-16:1-sn-glycerol-3-phosphate
1-18:3-2-16:3-monogalactosyldiacylglycerol
1-18:3-2-18:3-sn-glycerol-phosphocholine
1-18:3-2-trans-16:1-sn-glycerol-3-phosphate
1-chloro-2,4-dinitrobenzene
1-d-myo-inosityl-2-(l-cysteinyl)amido-2-deoxy-alpha-d-glucopyranoside
1-d-myo-inosityl-2-amino-2-deoxy-alpha-d-glucopyranoside
1-nitrate-1,2,3-propanetriol
1-phospho-2,3-diketo-5-methylthiopentane
1-phosphoryloxy-2,3-diketo-5-methylthiopentane
1-phosphoryloxyl-2-hydroxy-3-keto-5-methylthiopentene
13-[o(2')-beta-d-glucopyranosyl-beta-d-glucopyranosyloxy]docosanoate
13-[o(2')-beta-d-glucopyranosyl-beta-d-glucopyranosyloxy]docosanoate o(6)-acetate
13-[o(2')-beta-d-glucopyranosyl-beta-d-glucopyranosyloxy]docosanoate o(6'),o(6
13-sophorosyloxydocosanoate 6',6
13-sophorosyloxydocosanoate diacetate
2'-(5-phosphoribosyl)-3'-dephospho-coa
2'-(5-triphosphoribosyl)-3'-dephospho-coa
2,3-bis(3-hydroxytetradecanoyl)-d-glucosaminyl-1,6-beta-d-2,3-bis(3-hydroxy
        tetradecanoyl)-beta-d-glucosaminyl 1-phosphate
2,3-diketo-5-methylthio-1-phosphopentane
2,5,6-trihydroxy-3-methylpyridine
2,5-dichloro-cis,cis-muconic acid
2,6-dichloro-p-hydroquinone
2-(2'-methylthio)ethylmalic-acid
2-(alpha-lactyl)-thiamine-pyrophosphate
2-(alpha-lactyl)-thpp
2-(alpha-lactyl)-tpp
2-(sulfomethyl)thiazolidine-4-carboxylate
2-(sulfomethyl)thiazolidine-4-carboxylic acid
2-5-phosphoribosyl-3-dephospho-coa
2-amino-4-hydroxy-6-trihydroxypropyldihydropteridin-p3
2-amino-4-oxo-6-(erythro-1',2',3'-trihydroxyprolyl)-7,8-dihydroxypteridine
         triphosphate
2-amino-5-(fluoromethyl)pyrrolo[2,3-d]pyrimidin-4 (3h)-one
2-carboxy-3-keto-d-arabinitol-1,5-bisphosphate
2-chloro-trans-dienelactone
2-fluoro-l-erythro-citrate

2-hydroxy-3-keto-5-methylthio-1-phosphopentene
2-hydroxydeoxyadenosine 5'-triphosphate
2-ketoglutarate-dehydrogenase-system
2-nitrooxy-propane-1,3-diol
2-oxo-5-methylthiopentanoic-acid
2-phosphono-3-sulfoproprionate
20a-20b-dihydroxy-cholesterol
23-dihydroxybenzoylserine-multimers
2<l>trans</l>-2-chlorodienelactone
2fe-2s iron-sulfur cluster
3,4,6-trihydroxyphenylalanine quinone
3,4-dihydroxy-2-butanone-4-p
3,4-dihydroxy-2-butanone-4-phosphate
3,6-lactone of 3-hydroxyadinyl-coa
3-(2'-methylthio)ethylmalic-acid
3-benzoyloxypropyl-glucosinolate
3-dehydro-6-deoxoteasterone
3-enolpyruvyl-shikimate-3-phosphate
3-enolpyruvyl-shikimate-3p
3-hydroxy-4-phospho-hydroxy-alpha-ketobutyrate
3-hydroxy-5- oxohexanoyl-coa
3-hydroxypropyl-glucosinolate
3-methylthiopropyl-desulfo-glucosinolate
3-methylthiopropylhydroxamic-acid
3-oxo-delta4,6-cholyl-coa
3-oxo-delta4-deoxycholyl-coa
3-thiazol-2'-yl-indole
4-hydroxy-3-indolylmethyl-glucosinolate
4-methoxy-3-indolylmethyl-glucosinolate
4-nitro-5-aminoimidazole ribonucleotide
4fe-4s iron-sulfur cluster
5'-(p-nitrophenyl)thioadenosine
5,10-methenyl-tetrahydrosarcinapterin
5,10-methylene-5,6,7,8-tetrahydromethanopterin
5,10-methylene-tetrahydromethanopterin
5,10-methylene-tetrahydrosarcinapterin
5-(5-hydroxy-3-indolyl)-3-(3-oxinodolylidene)-2-oxopyrroline
5-formyl-tetrahydrosarcinapterin
5-hydroxy-3,4,4-trimethyl-delta2-pimelyl-coa-delta-lactone
5-methyl-tetrahydrosarcinapterin
5-methylaminomethyl-2-selenouridine
5-methylpyridine-2,3,6-triol
6-alpha-hydroxy-6-deoxocastasterone
6-alpha-hydroxycampestanol
7-alpha,12-alpha-dihydroxy-3-oxo-4-cholenyl-coa
7-aminomethyl-7-deazaguanine

7alpha,12alpha-dihydroxy-3-oxo-4-cholenoic acid
8-hydroxydeoxyguanosine 5'-triphosphate
9,10,18-trihydroxystearate
[1,4-(n-acetyl-beta-d-glucosaminyl)](n+1)
acetylmuramyl-alanyl-iso-glutamine
acetylmuramyl-alanyl-isoglutamine
adenosine(5')triphospho(5')adenosine
adenosine-3',5'-diphosphate malonyl-coa
adenosylcobalamin-5'-phosphate
alpha-d-galactosyl-(1,3)-beta-d-galactosyl-(1,4)-n-acetyl-d-glucosaminyl-r
alpha-d-glucosylpoly (glycerol phosphate)
amino-hydroxy-hydroxypropyl-dihydropteridin
another strange group
arf-gtp-arfreceptor
beta-d-gal-(1->3)-beta-d-glcnac-(1->3)-beta-d-gal-(1->4)-d-glc
c25-allenic-apo-aldehyde
c55-pp-glcnac-mannaca
c55-pp-glcnac-mannaca-fuc4nac
carnitine (d-form)
carnitine metabolism cofactor
carnitine metabolism pre-cofactor
cis-dihydrodiol derivative of phenylacetyl-coa
cyclohex-1,3-diene-5,6-dihydroxy-1-carboxyl-coa
dd-alpha-epsilon-diaminopimelate
delta2,5-3,4,4-trimethylpimelyl-coa
delta5,24-cholestadien-3-beta-ol desmosterol
delta7,24-cholestadien-3-beta-ol
delta7-cholesten-3-beta-ol lathosterol
deoxymorpholinofructose
dibenzothiophene-5-oxide
dihydroxybenzoylserine-iron
divinyl protochlorophyllide a
divinyl protochlrophyllide a
dolichol monophosphate mannose
erythrofluorocitrate (isomer)
ethyl-(2r)-methyl-(3s)-hydroxybutanoate
fe2s2 iron-sulfur center
fe3s3 iron-sulfur center
fe3s3 iron-sulfur cluster
fe3s4 iron-sulfur center
fe3s4 iron-sulfur cluster
fe4s4 iron-sulfur center
ferric dicitrate complex
ferric dihydroxybenzoylserine
ferric enterobactin complex
fes iron-sulfur center

fes iron-sulfur cluster
fructoselysine-6-phosphate
gamma-butyrobetainyl-coa
gdp-4-dehydro-6-l-deoxygalactose
gdp-4-keto-6-l-deoxygalactose
glcnac-pyrophosphorylundecaprenol
glycano-(2,6-alpha-n-acetylneuraminyl)-(n-acetyl-d-galactosaminyl)-
         glycoprotein
hydroxy-carboxypropyl tpp
hydroxydeoxocastasterone
hydroxylamine-o-acetic acid
indole carboxyl hydro-thiazole
indole carboxyl thiazole
indole-s-cysteine-adduct
indolylmethyl glucosinolate aglycone
indolylmethylthiohydroximate
indolylmethylthiohydroximic acid
inorganic open chain tetrapolyphosphate
iron-molybdenum cofactor
kaempferol-3-o-[glucopyranosyl(1-6)glucopyranoside]-7-orhamnopyranoside
kaempferol-3-o-gentiobioside-7-o-rhamnoside
kdo2-(palmitoleoyl)-lipid iva
kdo2-(palmitoleoyl-myristoyl)-lipid a
kdo2-lipid a, cold adapted
l-2-amino-3-oxobutyratex
l-alanyl-d-glutamyl-meso-2,6-diaminoheptane-d-alanyl-d-alanine
lipid intermediate i
lipid intermediate ii
mannaca-glcnac-fuc4nac-pp-lipid
mannaca-glcnac-fuc4nac-pyrophosphorylundecaprenol
mannaca-glcnac-pp-lipid
mannaca-glcnac-pyrophosphorylundecaprenol
methenyl-tetrahydrosarcinapterin
methyl-bis-3-chlorethyl
methylene-tetrahydrosarcinapterin
molybdopterin guanine dinucleotide
monoamide-of-a-dicarboxylic-acid
monovinyl protochlorophyllide a
monovinyl protochlrophyllide a
n-acetyl-beta-d-glucosaminyl-1,2-alpha-d-mannosyl-1,3-(n-acetyl-beta-d-
        glucosaminyl-1,2-alpha-d-mannosyl-1,6)-(n-acetyl-beta-d-glucosaminyl-
        1,4)-beta-d-mannosyl-1,4-n-acetyl-beta-d-glucosaminyl-r
n-acetyl-beta-d-glucosaminyl-1,2-alpha-d-mannosyl-1,3-(n-acetyl-beta-d-
        glucosaminyl-1,2-alpha-d-mannosyl-1,6)-beta-d-mannosyl-1,4-n-
        acetyl-beta-d-glucosaminyl-r
n-acetyl-beta-d-glucosaminyl-1,4-(n-acetyl-d-glucosaminyl-1,2)-alpha-d-mannosyl-

1,3-(beta-n-acetyl-d-glucosaminyl-1,2-alpha-d-mannosyl-1,6)-beta-d-mannosyl-r
n-acetyl-beta-d-glucosaminyl-1,6-beta-d-(n-acetyl-b-glucosaminyl-1,2)-beta-d-mannosyl-r
n-acetyl-beta-d-mannosaminyl-1,4-n-acetyl-d-glucosaminyldiphosphoundecaprenol
n-acetyl-d-galactosaminyl-1,4-beta-d-glucuronyl-1,3-beta-d-galactosylproteoglycan
n-acetyl-d-galactosaminyl-1,4-beta-d-glucuronyl-n-acetyl-1,3-beta-d-galactosaminylproteoglycan
n-acetyl-d-glucosaminyl-1,3-beta-d-galactosyl-1,4-n-acetyl-beta-d-glucosaminyl-1,3-beta-d-galactosyl-1,4-beta-d-glucosyceramide
n-acetyl-d-glucosaminyl-lipopolysaccharide
n-acetyl-d-glucosaminyldiphospho-undecaprenol
n-acetyl-glucosaminyl-pyrophosphorylundecaprenol
n-acetylmuramoyl-l-alanyl-d-glutamyl-l-lysyl- d-alanyl-d-alanine-diphosphoundecaprenol
n-acetylmuramoyl-l-alanyl-d-glutamyl-meso-2,6-diaminoheptane-d-alanyl-d-alanine-diphosphoundecaprenol
n-acetylmuramoyl-l-alanyl-d-glutamyl-meso-2,6-diaminoheptane-d-alanyl-d-alanine-diphosphoundecaprenyl-n-acetylglucosamine
n-acetylmuramoyl-l-alanyl-d-glutamyl-meso-2,6-diaminoheptanedioate-d-alanyl-d-alanine-diphosphoundecaprenol
n-benzoyl-d-arginine-1-4-nitroaniline
n-formyl-n-hydroxy-aminoacetate
n5,n10-methylene-5,6,7,8-tetrahydromethanopterin
n5-carboxyaminoimidazole ribonucleotide
rhizobactin 1021 core
s-(+)-3-hydroxy-4-trimethylaminobutyrate
s-(2-sulfoethyl)cysteine
succinate-semialdehyde-thiamine pyrophosphate
succinate-semialdehyde-thiamin pyrophosphate
sulfoquinovosyldiacylglycerol
tdp-4-acetamido-4,6-dideoxy-d-galactose
trans-3-hydroxycotinine-glucuronide
trans-delta2, cis-delta4-decadienoyl-coa
trans-delta2-decenoyl-coa
udp-4-amino-4-deoxy-l-arabinose
udp-l-ara4-formyl-n
udp-murnac-l-ala-d-glu-mesoa2pm
udp-murnac-tetrapeptide
udp-n-acetylmuramoyl-l-alanyl-d-glutamyl-meso-2,6-diaminoheptanedioate
udp-n-acetylmuramoyl-l-alanyl-d-glutamyl-meso-2,6-diaminoheptanedioate-d-alanine
udp-n-acetylmuramoyl-l-alanyl-gamma-d-glutamyl-meso-diaminopimelate
undecanyl-diphospho-glcnac-mannac-(p-gol)3(p-gol-glc)1
undecanyl-diphospho-glcnac-mannac-(p-gol)3(p-gol-glc)m
undecaprenyl n-acetyl-glucosaminyl-n-acetyl-mannosaminuronic acid pyrophosphate

undecaprenyl n-acetyl-glucosaminyl-n-acetyl-mannosaminuronic
   acid-4-acetamido-4,6-dideoxy-d-galactose pyrophosphate
undecaprenyl-n-acetyl-alpha-d-glucosaminyl-pyrophosphate
undecaprenyl-pyrophosphoryl-murnac-(pentapeptide)-n-acetylglucosamine
undecaprenyl-pyrophosphoryl-murnac-pentapeptide
uridine 5'-beta-l-threo-pentapyranosyl-4#NAME?
uridine 5'-pyrophosphoryl n-acetylmuramyl-tetrapeptide

Appendix E

Glossary

Activator: An entity that activates a binding reaction.

Binding Reaction: A reaction in which a set of reactants bind together to form a single product that is a complex of the reactants.

DNA-Binding-Sites: DNA regions to which transcription factors bind.

EST - Expressed Sequence Tag: A partial sequence of a randomly chosen cDNA, obtained from the results of a single DNA sequencing reaction. ESTs are used both to identify transcribed regions in genomic sequence and to characterize patterns of gene expression in the tissue that was the source of the cDNA.

Inhibitor: An entity that inhibits a binding reaction.

Operon: Sequence of genes responsible for synthesizing the enzymes needed for biosynthesis of a molecule. An operon is controlled by an operator gene and a repressor gene.

ORF - Open Reading Frame: A section of a sequenced piece of DNA that begins with an initiation (methionine ATG) codon and ends with a nonsense codon. ORFs all have the potential to encode a protein or polypeptide, however many may not actually do so.

Regulator: A gene that regulates the expression of one or more structural genes by controlling the production of a protein which regulates their rate of transcription.

Transcription Factor: A protein that binds to regulatory regions and helps control gene expression.

Appendix F

ACKNOWLEDGMENTS

Dr. Lenwood S. Heath

Mr. Allan A. Sioson

Dr. Ruth Grene

Dr. Naren Ramakrishnan

Dr. T.M. Murali

Department of Computer Science

Virginia Bioinformatics Institute

Mr. and Mrs. Rajasimha and family

Chaitanya, Satish, Maulik, Kiran, Nitin, Vyas, and all roommates and friends.

Appendix G

VITA

HARSHA K. RAJASIMHA is currently a research associate at the Virginia Bioinformatics Institute. He received a B.E. (2000) in computer science and engineering from Bangalore university, India and briefly worked as a lecturer in computer science in the same college. He gained industry experience interning as a software design engineer in test at Microsoft Corporation, Redmond, WA (Summer 2002). He worked as a graduate research assistant at the Virginia Bioinformatics Institute since December 2002 in the mitochondria research group. He intends to pursue doctoral research in bioinformatics. His research interests include biological data integration and modeling and simulation of life systems. He served as the secretary of the computer science graduate student council during the year 2003. He can be reached by email at <hrajasim@vbi.vt.edu> and his web address is <http://staff.vbi.vt.edu/hrajasim>.