Data mining and integration of heterogeneous bioinformatics data sources

Badr H. Al-Daihani Al-Mutairy

To my parents,
my wife,
and my brothers and sisters

Acknowledgements

My first and foremost thanks and praises are due to Allah (God) Almighty who has helped me and provided me with faith, patience and commitment to complete this research.

I would like to express my deep thanks and gratitude to my supervisor, Professor Alex Gray, for his supervision, guidance, support and encouragement throughout this research.

My special thanks also go to Dr. Peter Kille for his continued and unlimited help with regard to the biological aspects of my research. I am very grateful for his careful reading of, and constructive comments on this thesis.

Special thanks are due to the members of the school for their help, especially Mrs. Margaret Evans who has helped me with travel-related issues, Mrs. Helen Williams for her help in administrative issues, and Mr. Robert Evans and Dr. Rob Davies for their technical assistance.

I would also like to express my thanks to my fellow research students in the School of Computer Science at Cardiff University for providing a pleasant and stimulating research environment. I really enjoyed the friendship that I developed with them while doing this research.

Special admiration and gratitude are due to my parents, wife, brothers and sisters whose prayers, love, care, patience, support and encouragement have always enabled me to perform to the best of my abilities.

Last but not least, I would like to thank all the people, members of my family and close friends, who have borne with me during the period of my PhD studies.

Abstract

The integration of bioinformatics data sources is one of the most challenging problems facing bioinformaticians today due to the increasing number of bioinformatics data sources and the exponential growth of their content.

In this thesis, we have presented a novel approach to interoperability based on the use of biological relationships that have used relationship-based integration to integrate bioinformatics data sources; this refers to the use of different relationship types with different relationship closeness values to link gene expression datasets with other information available in public bioinformatics data sources. These relationships provide flexible linkage for biologists to discover linked data across the biological universe. Relationship closeness is a variable used to measure the closeness of the biological entities in a relationship and is a characteristic of the relationship. The novelty of this approach is that it allows a user to link a gene expression dataset with heterogeneous data sources dynamically and flexibly to facilitate comparative genomics investigations. Our research has demonstrated that using different relationships allows biologists to analyze experimental datasets in different ways, shorten the time needed to analyze the datasets and provide an easier way to undertake this analysis. Thus, it provides more power to biologists to do experimentations using changing threshold values and linkage types. This is achieved in our framework by introducing the Soft Link Model (SLM) and a Relationship Knowledge Base (RKB), which is built and used by SLM. Integration and Data Mining Bioinformatics Data sources system (IDMBD) is implemented as an illustration of concept prototype to demonstrate the technique of linkages described in the thesis.

Content

List of Figures

List of Tables

List of Acronyms

ACEDB	A Caenorhabditis Elegans Database
AcePerl	An object-oriented Perl interface for AceDB
API	Application Programming Interface
AQL	Acedb Query Language
BLAST	Basic Local Alignment Search Tool
BP	Biological Process
CAS	Chemical Abstracts Service
CC	Cellular Component
CDM	Common Data Model
cDNA	clone DNA
CPL	Collection Programming Language
DAVID	Database for Annotation, Visualization, and Integrated Discovery
DB	Database
DBMS	Database Management System
DBS	Database System
DDBJ	DNA Data Bank of Japan
DM	Data Mining
DNA	DeoxyriboNucleic Acid
EMBL	European Molecular Biology Laboratory
EC	Enzyme Commission
GO	Gene Ontology
GRAIL	GALEN Representation and Integration Language
GUI	Graphical User Interface
HMM	Hidden Markov Model
HTML	HyperText Markup Language

IC	Information content
IDMBD	Integration and Data Mining of Bioinformatics Data Sources
JDBC	Java Database Connectivity
JDOM	Java Document Object Model
JSP	Java Server Pages
MF	Molecular Function
MGI	Mouse Genome Informatics
OODBMS	Object Oriented Database Management Systems
ORDBMS	Object Relational Database Management Systems
OODM	Object Oriented Data Model
OQL	Object Query Language
OWL	Ontology Web Language
RC	Relationship Closeness
RDBMS	Relational Database Management System
RDF	Resource Description Framework
RKB	Relationship Knowledge Base
SAX	Simple API for XML
SEMEDA	Semantic Meta Database
SLA	Soft Link Adapter
SLM	Soft Link Model
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SRS	Sequence Retrieval System
TAMBIS	Transparent Access to Multiple Bioinformatics Information Sources
TaO	TAMBIS Ontology
PERL	Practical Extraction and Reporting Language
URL	Uniform Resource Locater

WM	Wrapper Manager
WWW	Word Wide Web
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Synopsis

Bioinformatics data sources are heterogeneous in their representation and query capabilities across diverse information fields held in distributed autonomous resources. The volume of data collected and stored in these distributed and heterogeneous data sources, presents a major challenge with respect to the efficient and effective accession, processing, extraction, discovery and integration of this information. In particular, this occurs when a biologist wants to use data mining tools linked with information held in existing knowledge and computational resources in investigations to exploit the exponentially increasing amount of comparative genomic data. In this chapter, a background to this problem is provided, followed by the research motivations for the thesis. Next, the hypothesis, the aims and objectives of the research are presented. The research methodology used is presented, followed by a summary of the overall achievements of the research. The chapter ends by describing the organization of the thesis.

1.2 Background to Integration of bioinformatics sources

The integration of bioinformatics data sources is one of the most challenging problems facing bioinformaticians today, due to the increasing number of bioinformatics data sources and the exponential growth of their content and usage [131, 138]. These sources usually differ in their structure, scope and contents [139]. Most data sources are centred on one primary class of objects, such as gene, protein, or DNA sequences. This means that each data source contains different pieces of biological information and knowledge reflecting the purpose of the source, and can answer queries appropriate to its domain, but cannot help with queries that cross domain boundaries and involve different data repositories. An area of research that is growing in importance.

In most existing integration systems, joining information held in different data sources is based on the uniqueness of common fields in the sources or by linkage through ontology terms. Data entries in some data sources have relationships expressed as links, or predefined cross-references. Such cross-references are usually stored as a pair of values, for example, target-data source and accession number, and are effected through a hyperlink on a webpage [36, 140]. These links are added to data entries for many different reasons: for example, data curators insert them as structural relationships between two data sources, and biologists insert them when they discover a confident relationship between items [36]. Yet, these links are not established in collaboration with the curator of the linked data sources. These static links (hyperlinks) are problematic, as the hyperlink may change. Thus, if a curator changes, or withdraws an entry that is related to an entry in another data source, the link fails [36, 140]. With sources changing quickly, this leads to inconsistency and continual updating is needed. Moreover, many bioinformatics data sources do not support explicit relationships with data held in other data sources, such as ortholog and other types of relationship. Bioinformatics data sources need linking using associations between entities that are hard to find, as they are implicit in the sources and not explicit in the data [3]. Relationships between data held in such data sources are usually numerous, and only partially explicit. There is, therefore, a growing need to link these data sources using dynamic and flexible linking at a higher level through relationships, particularly if this can be achieved in an efficient manner.

1.2.1 Experimental Datasets

The emergence of biotechnology has made it possible to study the expression of thousands of genes or proteins in a single experiment in the laboratory, which creates an experimental dataset [7, 181]. This raises many challenges:

- In order to mine relevant biological knowledge from an experimental dataset, it is important not only to analyse the experimental data, but also to cross-reference and associate the large volumes of data produced in this way with information available in external bioinformatics data sources, in order to conduct comparative genomics investigations and so predict gene functions and study evolutionary analysis [186].

- Due to the complexity of the biological problems under study and the lack of complete experimental and analytical models, there is a need to design a knowledge-driven system that assists in the explanation and validation of the predictive outcomes of experiments [198].

- Researchers have great difficulty in setting up large-scale experiments, mainly because of a shortage of expertise and limited resources to recruit appropriate staff [25], so most current researchers annotate genes one at a time, using online data sources or a manual literature search [106]. A previous study [107] has revealed that 40 to 60% of genes found in new genomic sequences do not have assigned functions.

- Many researchers struggle to identify the most appropriate sources and tools to be used in the analysis of their experimental datasets [106].

- One of the significant challenges is to integrate gene annotation with the gene expression and sequence information [136, 138, 193, 194], so that biologists can study genes based on their function, chromosomal location, and tissue expression, and cross-reference the data derived from different species across diverse expression analysis platforms.

- When linking and integrating data presented in an experimental dataset in a semi-structured form with data held in a bioinformatics data source, it is essential to determine as much information about the experimental dataset as possible. This information can be detected automatically from its metadata, such as column names and their content descriptions [75].

Thus, instead of overwhelming researchers with long lists of unannotated data, researchers need a system that allows them to annotate genes, and microarray[1] information by linkage to additional information from various online public data sources. The system should have the ability to integrate experimental datasets with the rich set of gene annotation information available within and across species. Such a system should allow researchers to collect and manage large amounts of gene expression, gene sequence, and gene annotation data.

In our research, we aim to develop a framework for integrating bioinformatics data sources that uses relationships across species and user preferences. It should allow the user to specify constraints and parameters for the integration, which would allow a biologist to facilitate flexible usage of different types of comparative genomics relationships in investigations.

1.3 Rationale

In 2006, over 100,000 individual samples were deposited in public repositories for gene expression/molecular abundance data. These submissions represent over 2000 platforms or array types from 60 different species [87]. This body of public data is growing

[1] Microarray is a high-throughput technology used in molecular biology and in medicine.

exponentially and is matched by an equal or greater number of studies in the private domain. Few tools have been developed to compare directly the results yielded from individual studies. Although, significant advances have been made in visualizing [22, 38, 47, 88] and manipulating individual datasets (including data processing [200], statistical analysis[103], clustering [16, 211] and annotation based over-representation [73]), these approaches allow only cross-experimental comparison by subjective analysis of the output. These comparisons offer an opportunity to reveal conserved disease mechanisms or common modes of action in cases of toxicosis caused by chemical exposure. The value of this data to the fundamental understanding of these processes cannot be underestimated, but new approaches are needed. The major hurdles to these dataset comparisons include variations in reported nomenclature, database versioning, orthology/paralogy, choice of relationship, and the threshold used to determine relationship validity. In this research, we set out to develop a platform that would allow direct comparison between two datasets, within species, allowing variable gene identifiers to be mapped onto the species-specific primary data source, which in turn could be used to yield sequence or gene annotation that would facilitate comparison, with flexibility in the types used and the thresholds of linkage.

1.4 The hypothesis and the aim of the research

The research hypothesis for this thesis is:

Hidden relationships between biological objects can be used in integrating bioinformatics data sources, so that a biologist can flexibly link an experimental dataset with bioinformatics data sources and the resulting data source can be mined effectively to inform the investigation.

Thus, the aim of the research is to investigate the use of relationships between biological objects to link heterogeneous bioinformatics data sources to annotate genes discovered in experiments and predict gene functions via comparative genomics analysis.

1.4.1 Objectives

In order to demonstrate the hypothesis, we aim to meet a number of objectives:

Objective 1: to extract an experimental dataset's metadata and to detect suitable candidate keys for linkage in it

Most experimental datasets are stored in unstructured files that do not have metadata saved in logical fields. In order to investigate fully the dataset being generated by a microarray or in a laboratory experiment, it is essential to detect and use as much information about the experimental dataset as possible. This information can be found in headings and content descriptions, and needs to be extracted and exploited to ensure that the data can be integrated in valid ways and so increase the scope of the investigations of the experimental dataset. Thus, a tool is needed to discover and extract this information.

Experimental datasets usually have many elements. Only a few of these elements can be used as a candidate key for linkage with other data. A candidate key helps us to join tuples in datasets with other data. Therefore, we need to try to detect automatically candidate keys that can be used to link and integrate a dataset with public data sources.

Objective 2: to transform extracted metadata and datasets into a form that can be used for linkage with other sources

Usually, experimental datasets are not in a form that can be directly linked to other bioinformatics data sources. The metadata should be stored in a format that allows its effective use. Also, datasets need to be analysed and stored so that they can be integrated and linked to other bioinformatics sources. Once the data has been stored in a suitable structure, it can be used to link with other appropriate public bioinformatics sources.

Objective 3: to show that these relationships can provide flexible and loosely coupled linkages across heterogeneous data sources

Bioinformatics data sources contain a large variety of objects. These objects are connected in a variety of ways giving an extensive interconnected graph of relationships. These relationships are often many-to-many, and refer to dynamic effects that one object has on another. Discovering these relationships between biological objects is important for biologists so that they can investigate whether the links enrich their knowledge about the genetic structure. Thus, the discovered relationships provide a means for joining information and linking data sources dynamically and flexibly, and so provide biologists with rich information and annotation. Thus, the objective is to detect these semantic relationships and build a relationship knowledge base containing this information that can be used to join information based on the GO classification association or homology between sequences, so that a biologist can assess the significance of the different links used in an investigation.

Objective 4: to build a knowledge base of discovered relationships between sources and to exploit this to combine annotation knowledge from different sources.

Discovered relationships between biological objects will be stored in a knowledge base that can be used in the integration process to enrich a query. User queries can be extended using these relationships to obtain a greater amount of relevant information. The objective is to store these relationships in an appropriate model so that they can be reused in future investigations.

Objective 5: to provide users with uniform access to bioinformatics sources so that they can be queried as if they were a single source, thus shielding users from the underlying structure of sources.

An integration aim is to provide users with a single interface to access and query multiple bioinformatics sources. The system should enable users to submit a single query to multiple bioinformatics data sources, and return a unified set of results rather than the user having to spend unnecessary time submitting the same query over and over again to many data sources and then integrating the results manually. Moreover, end users of the integration system should not need to be aware of the underlying structure of sources when accessing or querying heterogeneous data sources. The system should handle all the underlying mechanics needed to process a user's query and return results. The objective is to hide the internal structure of these sources from users to simplify the interface for the biologist.

1.5 Research Approach

In this section, we summarise the methodology used in conducting our research. Firstly, the problem is defined as linking experimental datasets from biological experiments with heterogeneous bioinformatics data sources in flexible ways to support knowledge discovery, comparative genomics, or further investigation. Existing integration systems are then reviewed to determine the most appropriate approach. The literature review is split into two tracks; the first concentrates on the integration of heterogeneous data sources in general and the second is about bioinformatics data source integration and the mining of biological data. These tracks are then combined to support the research aim.

Discussions with professionals in biological science was undertaken, as it was our targeted application field. Dr. Peter Kille (*Bioscience School, Cardiff University*) was frequently consulted to ensure that our research met a biologist's needs. Experimental datasets were collected under the supervision of staff of the School of Bioscience. Different bioinformatics data sources were selected to be integrated with these datasets based on the biology under investigation, namely, Wormbase [46, 210], MGD [33-35, 41, 71] and Gene Ontology (GO) [89].

Based on our investigation of the research problem, we built a model for capturing and storing relationships between the biological objects to be used for the integration and linkage of the bioinformatics data sources. An initial system structure was proposed which provided a user with uniform access to heterogeneous bioinformatics sources. The final step in our research was the implementation of our proposed system as a prototype.

1.6 Overall Achievements of the research

The following is a summary of the main achievements of this research:

a) Introducing an approach for extracting an experimental dataset's metadata and identifying appropriate candidate keys for linkage with other related data (Chapter 6).

b) The creation (see Chapter 4) of a novel approach – SLM - to the integration of bioinformatics data sources which allows biologists to create easily, different types of linkages between bioinformatics data sources, drive the integration process, change the linkage type flexibly, adjust the linkage easily, so that the investigator can try different linkages, see the effect of using them and so determine which one if any matches the purposes of their research and produces significant results. This allows biologists to analyze experimental datasets in different ways, shortens the time needed to analyze the datasets and provides an easier way to undertake this analysis. Thus, SLM provides biologists with a tool which supports experimentation by using different threshold values and linkage types and thereby supports investigative research (Chapter 8).

c) The creation of a knowledge base of the discovered relationships between biological objects (Section 9.4), which is used to compare and link the experimental datasets with public sources. This knowledge base improves comparative approaches to annotate genes, by identifying possible relationships between objects across species, and predicting protein-function from sequence homology, orthology and GO-terms. By integrating functional and sequence data across species, biologist can annotate the genome of a species using functional data from another. Comparative genomics provides evidence for close evolutionary relationships between gene families. Also, this knowledge can be reused in other investigations.

d) A flexible mediator architecture for linking (i.e. integrating) experimental datasets with relevant information held in heterogeneous data sources (see Chapter 5). This means that a biologist does not need to directly query individual data sources or use a variety of Internet search tools for this purpose. We present a mediator-based integration architecture that links experimental datasets to relevant information held in heterogeneous data sources. Our mediated architecture offers a set of tools for discovering semantic relationships between biological objects, browsing these relationships and automating metadata extraction, and offering a single point of access to a set of data sources. It enables flexible integration of heterogeneous data sources. This allows biologists to be able to create easily, different types of linkages between bioinformatics data sources, drive the integration process, change the linkage type flexibly, adjust the linkage easily so that the investigator can try different linkages to see which one if any matches the purposes of their research and determine the effect of different relationships easily and so identify their biological significance.

e) The Determination of the optimal threshold for cross-species orthology relationships. This is demonstrated for Mouse and C.elegans (see Section 8.5).

Six papers were published on the work reported in this thesis. The full details of these papers are found in [8-12]. The conferences and the workshops in which the papers appear are:

1. 21st Annual British National Conference on Databases, BNCOD 21, Edinburgh, UK, 7-9 July 2004.
2. Sixth Informatics Workshop for Research Students, University of Bradford, Bradford, UK, March 2005.
3. 22nd British National Conference on Databases, BNCOD 22, Sunderland, UK, 5-7 July 2005.
4. HIBIT 05: International Symposium on Health Informatics and Bioinformatics, Belek, Antalya, Turkey, 10-12 November 2005
5. 4th International Workshop on Biological Data Management - BIDM '06 in conjunction with DEXA 2006, Krakow, Poland, 3-7 September 2006.
6. VLDB 2006 on Data Mining in Bioinformatics in conjunction with VLDB 2006, Seoul, South Korea, 11-15 September 2006.

1.7 Thesis organization

This section presents an overview of the thesis organization. An overview of the chapter contents is given.

- ***Chapter 2: Background***
  This chapter gives the necessary background information about the characteristics of biological objects and bioinformatics data sources.

- ***Chapter 3: Bioinformatics Data source Integration***
  This chapter surveys the background areas of research related to the main ideas presented in the thesis on linking datasets.

- ***Chapter 4: Soft Link Model***

This chapter introduces the proposed Soft Link Model for data source integration and describes the approach used.

- ***Chapter 5: System Architecture***

This chapter introduces the design of the architecture and the different components of the IDMBD (Integration and Data Mining of Bioinformatics Data sources) system.

- ***Chapter 6: Implementation***

This chapter discusses the implementation issues for the proposed system, and describes the prototype implementation.

- ***Chapter 7: Extracting Metadata of Experimental Dataset***

This chapter presents an approach for extracting the experimental datasets' metadata and finding the suitable linkage keys that can be used for integration based on a mathematical foundation. Furthermore, it shows how to map a linkage key with the domain ontology to find related concepts and semantic relationships.

- ***Chapter 8: Analysis of "wet laboratory" data***

This chapter demonstrates the utility of our prototype system. We used the tools to analyse datasets generated by wet laboratory experimentation. The aim was to demonstrate that the soft link framework would allow us to derive novel insights into the experimental system by determining the elements conserved between species.

- ***Chapter 9: Evaluation***

This chapter provides an evaluation of the system in terms of different dimensions.

- ***Chapter 10: Conclusions and future work***

This chapter summarizes and comments on the contributions made by the research and discusses the perspectives and research directions that remain open for future work that could be carried out to improve the effectiveness of the SLM as a method of integrating heterogeneous bioinformatics data sources.

CHAPTER 2

Background

2.1 Synopsis

This chapter gives the background about biological data and bioinformatics data sources. The necessary background information about bioinformatics data sources is presented. This covers reasons for the growth in the number and size of bioinformatics data sources, and the characteristics of bioinformatics and its data sources. This growth is often described in the literature as explosive[113, 187, 214]. Heterogeneity present in bioinformatics data sources is detailed and types of conflict explained. Data models are defined and described in detail, and their advantages and disadvantages discussed.

2.2 Introduction

In recent years, there has been a massive increase in the number and size of bioinformatics data sources, which is expected to continue at the same, or an even faster pace in the coming years [131]. The growth in the number of data sources is related to the content of data held in them [65]. The reasons for this growth can be summarised as follows:

i. Rapid progress of the human genome project and other sequencing projects [58];
ii. Easy access to stored data provided by the Internet [13, 131];
iii. Proliferation of new biodata analysis technologies, bio-statistical approaches, computational algorithms, knowledge discovery, data mining and data analysis tools [60, 157];

iv. Design and development of new biotechnology and efficient (with respect to speed and accuracy) experimental techniques, primarily DNA sequencing, DNA microarrays and other high throughput technologies [131]; and

v. Massive investment in genomics by governments and the pharmaceutical industry [92, 131, 199].

In June 2008, the GenBank database alone held the records of more than 88,554,578 sequences and over 92,008,611,867 bases [86]. According to a recent survey, more than 1078 bioinformatics data sources are available online [83]. Table 2.1 and Figure 2.1 show the increase in the number of bioinformatics data sources from 1999 to the present day. Figure 2.2 illustrates the development of the international Nucleotide Sequences database [86]. Figure 2.3 shows the growth of the GenBank database from 1982 to 2005. In this period, there was an exponential growth in base pair data from 680K to 56,037 million and in sequences from 606 to 52 million [85]. Such explosive growth is expected to continue well into the 21st century [113, 114, 187, 196].

Data sources are maintained by different communities and organizations [131, 138]; they are autonomous, distributed, disparate, heterogeneous and often do not provide direct access [29, 138]. A description of these characteristics can be found in section 2.3.2.

Data sources in general can be classified as primary or secondary. A primary source holds information from an experiment and is sometimes called an archival data source. It contains raw data of sequences or structures. Examples of these primary sources are GenBank [31, 32], EMBI and DDBJ for Genome sequences and the Protein Databank for protein structures [21].

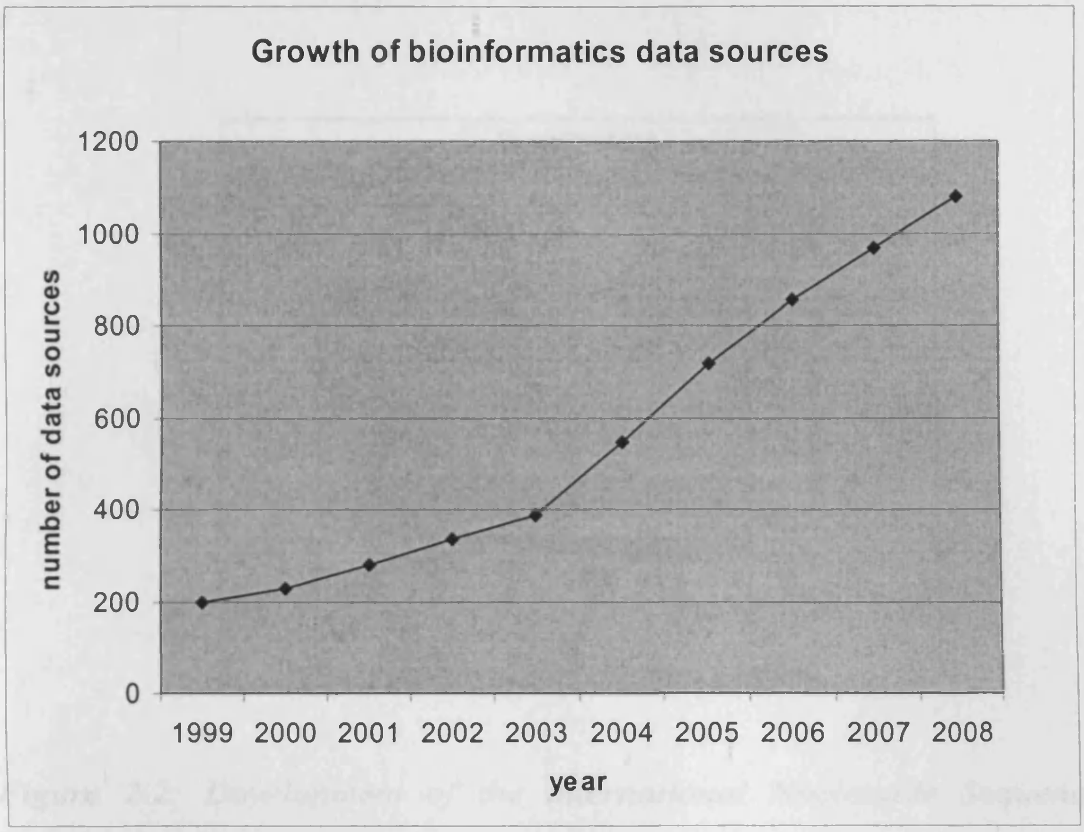


Figure 2.1: Growth of bioinformatics data sources 1999-2008 based on statistics published in [79-83]

Year	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
Number	197	226	281	335	386	548	719	858	968	1078

Table 2.1: Growth of bioinformatics data sources (1999-2008)[82-85]

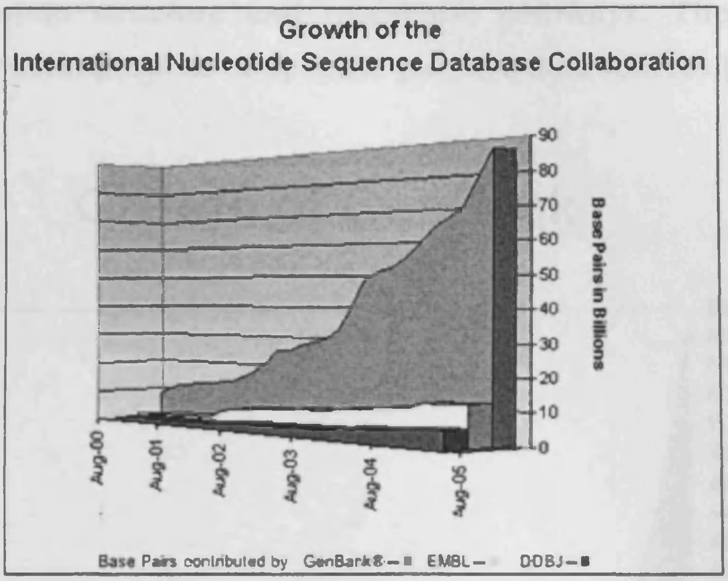


Figure 2.2: Development of the international Nucleotide Sequence Database [85]

Secondary data source information is derived from primary data source data; Secondary data sources hold data, such as conserved sequences, signature sequences and active site residues of the protein families derived by the multiple sequence alignment of a set of related proteins. A secondary data source is called a curated data source and examples include MGD [34] and Wormbase [46].

While the contents of primary data sources are controlled by the submitter, the contents of secondary data sources are controlled by a third party. Secondary data sources are derived from the following procedures [132]:

- Annotating and enriching data, either manually or automatically,
- Cleansing and removing redundant information,
- Collecting data from literature,
- Mining and compiling data from several data sources, and
- Analysing primary data.

In general, bioinformatics data sources cover a wide range of subjects and data types, including gene sequences, gene expression data, protein sequences, protein structure and metabolic pathways. They can be classified as general purpose or specific purpose data sources [29].

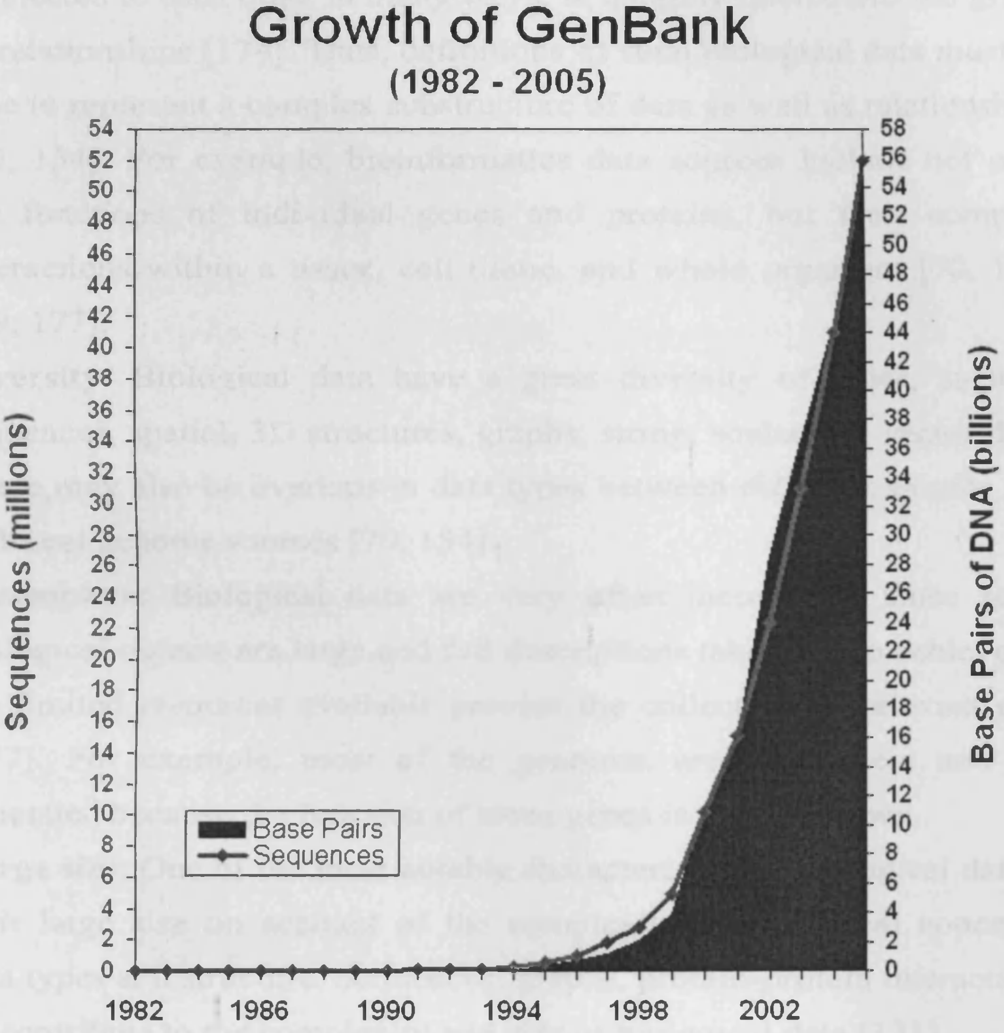


Figure 2.3: Growth of GenBank (1982-2005) [85]

2.3 Characteristics of bioinformatics data sources

The characteristics of bioinformatics data sources are presented here to give the reader an understanding of the field and the challenges it presents.

2.3.1 Data

Elmasri and Navathe [70] identify several characteristics of biological data that make it difficult to manage:

Complexity: biological data are questionably the most complex data known when compared with most other applications [177]. They are connected to each other in many ways, in a highly interconnected graph of relationships [174]. Thus, definitions of such biological data must be able to represent a complex substructure of data as well as relationships [70, 154]. For example, bioinformatics data sources include not only the functions of individual genes and proteins, but their complex interactions within a tissue, cell tissue, and whole organism [70, 154, 159, 177].

Diversity: Biological data have a great diversity of types, such as sequences, spatial, 3D structures, graphs, string, scalar and vector data. There may also be overlaps in data types between different species and different genome sources [70, 154].

Incomplete: Biological data are very often incomplete since some biological objects are large and full descriptions take time to achieve, or the limited resources available prevent the collection of relevant data [177]. For example, most of the genomes are incomplete and not annotated because the function of some genes is still unknown.

Large size: One of the most notable characteristics of biological data is their large size on account of the complexity of biological concepts, data types and structure. Sequences, graphs, protein-protein interactions all contribute to the complexity and size of biological data [131].

Lack of a standardised nomenclature: Different organisations and communities use their own terminology to describe biological concepts. Thus, biological data frequently suffer from ambiguous and unclear concepts since there is no standardised nomenclature for them [131, 177].

2.3.2 Data sources

Here we discuss the differing characteristics of bioinformatics data sources [29]:

Heterogeneous in structure and content: each data source has its own data model and uses its own terminology and ontology. Different designers, have used several ways to model a particular concept and the aim of the experiment and project all contribute to this heterogeneity [98, 154]. Thus, the structure of data sources, and representations of the same data query results may be different (see section 2.4).

Large in size: in the last few years, the number and size of new bioinformatics data sources has been growing exponentially, as has the number of computational tools available for analysing these data. There is no sign of any deceleration of growth [29].

Dynamic: bioinformatics data sources are dynamic. Their interfaces alter from time to time and their schemas change at a rapid pace as do their contents [70].

Autonomous: bioinformatics data sources are autonomously owned and maintained by different communities and organisations often for different purposes [138]. Consequently, query types allowed on data sources and the precise mode of interaction are diverse because of the different reasons for holding the data [29, 138].

Widely distributed: bioinformatics data sources are widely distributed across the world, and such data is currently not held in a centralised location for analytical purposes. This is most likely to continue to be the case [29, 138].

2.4 Heterogeneity in Bioinformatics Data Sources

This section identifies different types of heterogeneity that affect bioinformatics data sources with the aim of showing the challenges they present to making an interoperable system. This heterogeneity may exist at three levels, namely, syntactic, semantic and data model levels [26, 69, 84, 99, 110, 123, 128, 129, 131].

2.4.1 Syntactic

Syntactic conflicts, some referred as technical conflict, arise due to the use of different storage paradigms and formats, platforms, type of systems and communication protocols [128, 131, 134]. Syntactic conflicts may also occur due to the use of different query interfaces, for example, SQL, OQL, Xquery/Xpath, the access method used, for example, ODBC, JDBC, SOAP, and the storage method [128, 131].

2.4.2 Semantic

The classification of the semantic heterogeneities can be found in [69, 98, 99, 128, 129, 131].

Won Kim [128] describes a schema as

> *"containing a semantic description of the information in a given database, which can be represented in many ways in the same data models. Given such inter- and intra-model variability, it is a formidable task to integrate many schemas into a homogeneous schema."*

Thus, semantic conflicts are concerned with differences in the representation, meaning, interpretation or use of the same or related data [26, 84, 98, 99]. The most important semantic heterogeneity affecting bioinformatics data source integrations are:

Schema conflicts: concepts may be represented using different data structures in different databases, for example, an entity in one schema may be an attribute in another, different data types are used (string or integer), different units are used (pound, kilo), and the precision may vary (two or four decimal places; mark or grade of a metric). Other causes of conflict include different ways of representing incomplete information (for example, the meaning of nulls), and different ways of identifying objects in databases [69, 98, 99, 152]. Another conflict is data value conflict; this arises when different representations are used for equivalent data. These conflicts include discrepancies of type, unit, precision, allowed values, spelling and abbreviations [98, 99, 152]. For example, gene number is represented by Arabic numbers in MGI and Roman numbers in Worm.

Data versus schema conflicts: these conflicts arise when data (values) in one schema are considered as metadata (type names) in another data source. For example, a data value in one relational schema may be the name of an attribute in another relational schema [98, 129, 152].

Entity identification conflicts: entity identification conflicts occur when there is difficulty determining whether two or more entities (instances) in different data sources refer to the same real world entity. For example, a mouse gene identifier in MGI is different from the same gene accession number in Genbank [152].

Naming conflict: naming conflicts arise when different names are used for the same concepts in the real world or the same names are used for different concepts in the real world. This occurs when the designers' terminology and nomenclatures used to describe a real world concept lead to synonym and homonym problems. In the first, two different names are used to describe the same concept; for example, some data sources use common English species names while others use systematic species names. In a homonym, the same name is used to describe different real world concepts [98, 99, 152].

Generalisation/Specialisation Conflict: some protein domains have functional annotations from different sources. Thus, generalisation/specialisation conflict may occur. For example, sources may describe the same gene function using the gene ontology molecular function but use different hierarchical levels [152].

Linked Conflict: this is caused by the method used to link sources. For example, MGI links to Swiss-Prot through its marker concept, to RatMap through orthologs, to PubMed through references, and to GenBank through their markers (for genes) or molecular probes and segments (for anonymous DNA segments) [138].

Scope conflict: this arises when one source clearly encodes the scope of its data with respect to species, whereas another source refers to the species implicitly, as it covers only one species [131, 134, 135].

2.4.3 Data models

A data model is an abstract, logical definition of the *objects* used to model the structure of data [55-57, 184]. Data model conflicts occur when databases use different models, for example, relational, object-oriented, AceDB, hierarchical, to model the data [98, 146].

Flat files: it is estimated that 80% of biological data are in text form [191]. In the past, bioinformatics data were normally stored in ASCII text files. Today, many bioinformatics data sources are held in flat files, which are a single, large table, containing only one record structure and no links between separate records. This flat file is structured using letter codes at the beginning of each line [40]. Access to data in flat files is carried out sequentially, so access is slow because the entire file must be searched sequentially to find the wanted data. They also suffer from data redundancy, inconsistent data, inflexibility, limited data sharing, poor enforcement of standards, low programming productivity, and excessive program and data maintenance [141].

Currently, there is a shift to hold bioinformatics data sources in relational, object or object relational database management system (DBMS) or as XML data. Flat files are no longer considered appropriate alternatives to DBMSs. However, flat files are the de facto data exchange standard in the field, since many bioinformatics applications operate on flat files, for example, BLAST[15] and FASTA [143].

ACeDB: ACeDB is a database management system developed to store data of a small worm called C. elegans. In [5] it is described as follows:

> *"ACeDB was originally developed for the C. elegans genome project, from which its name is derived (A Caenorhabditis elegans DataBase). However, the tools in it have been generalised so as to be much more flexible and the same*

> *software is now used for many different genomic databases from bacteria to fungi to plants to man. It is also increasingly used for databases with non-biological content."*

Thus, ACeDB can refer to a database and data relating to the nematode C. elegans, or to this database management system. Only a few, but nevertheless significant, bioinformatics data sources are implemented using ACeDB [40]. The AceDB model has several advantages – accommodation of rough data items; easy extension of the schema; and a powerful and high level query language called AQL; furthermore, it is an appropriate model for small to medium sized internal databases [40].

Object Oriented Data Model: the Object Oriented Data Model (OODM) evolved in the mid-to-late 1980s subsequent to the appearance of object-oriented programming languages, such as C++ [126]. According to Bry and Kroger [40], in 2003, about 7% of all molecular biological databases are implemented using Object Oriented Database Management Systems (OODBMSs). A clear advantage of the OODB is its ability to represent the relationships between biological objects. Moreover, complex data types that can be implemented using object oriented programming language can be stored by storing objects.

Relational Data Model: the relational data model was first introduced in 1970 [50]. A relational model represents data as a two-dimensional table called a relation. It is based on the mathematical theory of relational algebra and calculus [56]. Since a considerable amount of bioinformatics data sources are based on proprietary flat file solutions, relational DBMSs are not as popular for bioinformatics data sources as in other application domains, for example, business applications. Recently, many flat file data sources have been converted to relational DBMSs [40]. Searching, analysing, and comparing sequences is not possible within relational databases, although some systems have recently been developed that facilitate sequence analysis. The relational model does not support all types of relationships between biological entities in a direct and intuitive way [141, 167].

Object-Relational Data Model: Stonebraker [184, 185] and Kim [78, 127] developed the object-relational data model (ORDM) in the 1990s. The ORDM has inherited the robust transaction and performance management features of the relational model and the flexibility of the object-oriented data model. According to Bry and Kroger [40], about 3% of all bioinformatics data sources are implemented on Object Relational Database Management Systems (ORDBMS).

The issue of the interoperability and integration of bioinformatics data sources has received considerable attention in bioinformatics. Many bioinformatics integration systems have been developed (Chapter 3). Interoperability is required since it is not practical to build a single database for all biological data. Most of the conflict resolution techniques used in bioinformatics can be found in [61, 128].

2.5 Summary

This chapter introduced the necessary background about biological data and bioinformatics. It covered the growth of biological and bioinformatics data sources. Then it highlighted some characteristics of biological data and sources and challenges of integration. Finally, it classifies the heterogeneity present into types of heterogeneity. In the next chapter, we will discuss different integration approaches and survey some of the existing bioinformatics integration systems.

Chapter 3

Bioinformatics Data Source Integration

3.1 Synopsis

In this chapter, general approaches to integrating heterogeneous bioinformatics data sources are discussed and each approach is described briefly. Several bioinformatics data source integration systems that have been reported in the literature are then surveyed, leading to the presentation of the framework of our approach.

3.2 Introduction

Bioinformatics data sources are heterogeneous in their representation and query capabilities across diverse information fields, and are held in disparate, distributed, autonomous data sources [138, 139]. The volume of data collected and stored in these distributed and heterogeneous data sources presents a major challenge with respect to efficient and effective accession, and the processing, extraction, discovery and integration of this information [209]. Using existing knowledge, computational resources and data mining tools, a biologist can exploit the exponentially increasing amount of comparative genomic data to formulate novel hypotheses [195], leading to the informed design of new cycles of laboratory research [138, 209]. There are several ways of testing such hypotheses, which are effective when data is static and standard linkage types are to be used, but limited when the data is dynamic or novel types of linkage are required. These limitations are caused by the evolving and changing nature of the data in these data sources, which means the researchers need to work with the most up-to-date version of the data and be able to utilise different linkages in the investigations. These changes in the data sources are due to the evolving understanding of the field where new gene annotations are continually being discovered and the findings from new bioinformatics investigations lead to new knowledge. This means that there is a need to update the data held in the data sources to reflect the new understanding [209].

In order to perform a high-throughput analysis of biological data, it is necessary to access and process information from a variety of data sources using standard and proprietary query interfaces and analytical tools. These data sources may be heterogeneous, distributed over intranets or the Internet, or may exist in a large number of public biological data repositories and require diverse applications to access, filter, interpret and combine them.

3.3 Integration approaches

Integration approaches can be classified according to the architecture and integration strategies used (see Figure 3.1). The linkage can be achieved using one of the three types of strategy (see Figure 3.2).

3.3.1 Architecture

Data integration and the linkage of bioinformatics data sources have attracted the attention of researchers for several years [4, 64, 119, 131]. Existing systems for integrating bioinformatics data sources use a number of different integration approaches. Currently, there are four basic models: mediation, federation, warehousing and navigation or link-based integration (see Figure 3.1).

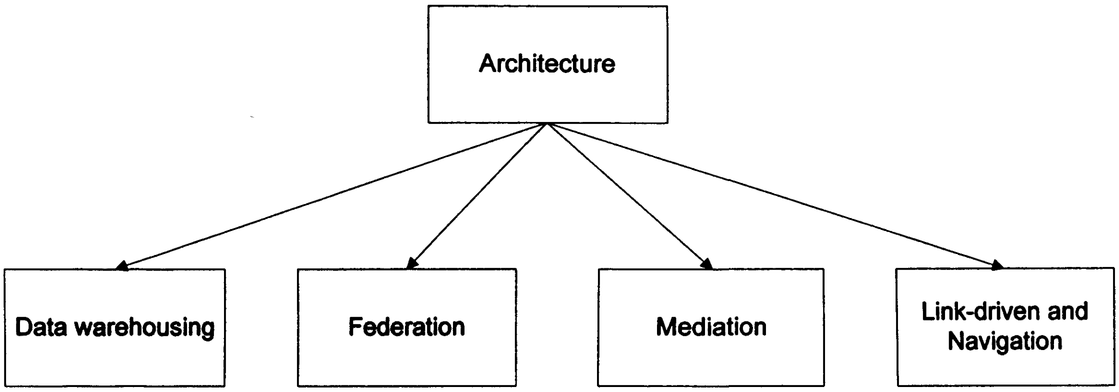


Figure 3.1: Basic data integration models based on architecture

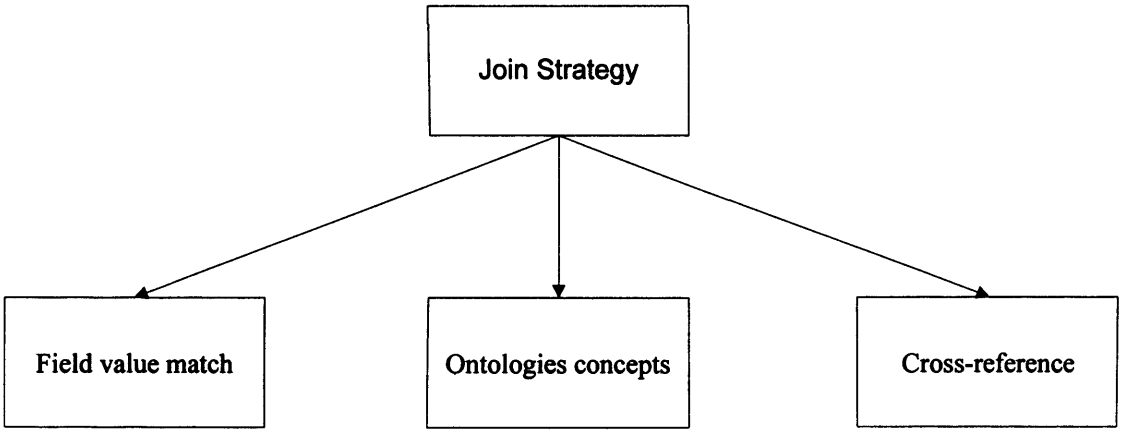


Figure 3.2: Basic joining and integration strategies

3.3.1.1 Data warehousing

Data warehousing brings data from different data sources into a centralised local system so that they can be integrated and shared [138]. Data warehouses often use wrappers to import data from remote sources. These data are materialized locally through a global schema used to process queries. While this simplifies the access and analysis of data stored in heterogeneous data repositories by bringing them to a central store with a common structure, the challenge is to keep the data in the warehouse current when changes are made to the remote sources. This is a particularly difficult task when the warehouse is large and the sources being linked are disparate, widely dynamic and autonomous. It requires a large maintenance effort and an in-depth understanding of data schema. On the other hand, data can be readily accessed, without delay or bandwidth limitation, and duplication, errors and semantic inconsistencies can be removed through applying data warehousing procedure.

The main advantages of this approach are that system performance tends to be much improved. Query optimization can be performed locally and communication latency to access various data sources is eliminated. System reliability is also improved since there are fewer dependencies on network connectivity and the availability of the data sources. Another advantage is that, while the underlying data sources may contain errors, a separate cleansed copy of correct data can be kept. Moreover, the researchers can add additional information, or annotation, to this data, which can be significant. However, because a warehouse requires a large maintenance effort as the underlying data sources change, this generates several practical problems, such as how to detect whether the remote sources have changed, how to automate the refresh process, and how to track the origins or 'provenance' of data [59]. In addition, the complexity and cost of maintenance can make large scale data warehousing impractical for large biology laboratories [131]. This approach might be realistic only at a moderate scale when dealing with a limited set of data sources [59]. Thus, the cost of the maintenance, storage and updating of data are critical issues in data warehousing. DAVID [63], GUS [95], and AllGenes [138] are examples of this approach.

3.3.1.2 Federation

In a federation architecture, the database systems are independent and autonomous [179]. Data are accessed from their original location and retrieved via a middleware component, which uses a common data model and a mapping schema to map heterogeneous data source schemas into the integrated schema. While this approach provides users with up-to-date data by accessing the local data source, the maintenance of a target schema can be costly due to frequent changes in data source schemas. Moreover, complete understanding of all the individual data sources is required and each source has its own wrapper, which must be maintained by the federation [96, 179]. Also, since data in data sources may not be clean, integrating dirty data may generate integrated dirty data or cause complications in the integration process. Thus, significant overheads may be needed to connect heterogeneous data sources, execute a user query, receive data from sources, merge data into a single result set, and return a result to a user. The main advantages are that it preserves source autonomy and uses the most recently available version of data. K2/Biokleisli [58, 59, 138] and DiscoveryLink [97, 138] are examples of this approach.

3.3.1.3 Mediation

In 1992, Wiederhold introduced the mediator-wrapper architecture [201], which has an intermediate processing layer called the mediator and decouples the data sources and client layers. This mediator offers an integrated view of data sources through wrappers. The mediator provides a virtual view of the integrated sources that is read-only. The mediator interacts with the autonomous data sources via wrappers, and handles a user query by splitting it into sub-queries, sending the sub-queries to appropriate wrappers, and integrating the results locally to provide responses to queries. Examples of mediation systems are covered in [45, 134, 135, 138, 139].

3.3.1.4 Link-driven and Navigation

This architecture is widely used in Web retrieval. Many data sources provide links to other data sources. Usually, accession numbers or other global identifiers are used for interlinking, for example, The Life Science Identifier (LSID)[2] [48]. Some databases use other attributes for the interlinking, such as ontology terms [18], EC numbers [2] and CAS registry numbers [42]. However, as different data sources use different identifiers for the same entries, it is a labour-intensive approach. For this reason, most databases provide links only to the most relevant databases via accession numbers [131, 133-135]. Examples include the Sequential Retrieval System (SRS) [74, 138], BioNavigator and Entrez [145, 175]. Since this type of integration system allows users to navigate from one source to another via predefined static links, there is a limit on the scope of user queries. Other drawbacks are that links are static and unidirectional, may not exist between related entries or may have been broken or have poor scalability; furthermore, usually there are no common keys to join tables and data sources. As bioinformatics data sources have different formats, such as flat files, XML, HTML, unstructured, relational and object-oriented files, cross-referencing does not always work in a straightforward way [131, 134].

3.3.2 Joining and matching strategies (mechanism)

In this section, we describe methods used to link different data sources together in different approaches. This linkage can be achieved using one of the three types of strategy (see Figure 3.2).

[2] An LSID is represented as a Uniform Resource Name (URN) with the following format.

URN:LSID:<Authority>:<Namespace>:<ObjectID>[:<Version>]

3.3.2.1 Integration based on matching keyword values (keyword-based)

The most familiar approach for integrating data is to match fields between data sources. For example, two entries from diverse data sources may be linked based on the identity of an accession number in these entries. Identifiers (accession numbers) are often used to join, interlink and integrate. However, this simple matching strategy may not give high quality integrated data, due to semantic heterogeneity between sources. In brief, different sources may use different terminologies. For example, one source may use scientific names for a species (Mus musculus or Escherichia coli) while another uses the common name (mouse or Bacterium coli). In addition, even when data sources use the same terminology, different lexical variants may be used for the same term, for example, "B Cell leukaemia", "Leukaemia, B Cell" or "B-Cell Leukaemia's". Further, the resolution level of the data may differ across sources. For example, one source may describe a disease phenotype as "Leukaemia" while another specifies "Leukaemia, B-Cell, Acute" [44]. Examples of systems using this approach are SRS [74, 138], and Entrez [172, 175].

Thus, a common approach is to integrate information based on syntactical equivalence, i.e., two objects with the same name (or two fields with same value). However, this is not always sufficient because names of biological objects (proteins, genes, pathways) are sometimes assigned by different laboratories in different communities and so differ; thus, other approaches based on the characteristics of objects are needed [36].

3.3.2.2 Usage of ontology (concept-based)

According to Gruber's definition [93], "*an ontology is a specification of a conceptualisation*". An ontology is the formal specification of vocabularies of concepts and the relationships among them in a domain. Use of an ontology in data source integration has previously been studied by [51, 62, 162]. An ontology also plays a role in heterogeneous data source integration in which the terms are mapped semantically to a concept on a proprietary ontology [176]. A survey on the use of ontologies for heterogeneous database integration can be found in [176]. An ontology can be used to support the integration of data from different external data sources in a transparent way, capturing the exact proposed semantics of the data source terms, and removing mistaken synonyms. The domain ontology of systems like TAMBIS[23, 182] and SEMEDA [134] allows users to formulate queries without knowledge of the underlying data source or direct access to the sources [176]. This means that the users do not need to know the underlying structure of data sources.

An ontology can help in solving interoperability problems among heterogeneous databases, since it establishes a common understanding of the terminology between different research communities. It provides definitions for the vocabulary used to represent knowledge and can be used to create an integrated schema that provides specific and complete models of particular domains [17].

In recent years, ontologies have been widely used for database integration and searching [24, 43, 165, 197]. Different ontologies and approaches have been used in the domain of bioinformatics. Some integration systems use a single ontology approach and others use multiple ontologies for integration purposes. A single ontology approach can be used to support integration when the sources share nearly similar views on a domain [54]. However, if the bioinformatics sources have different views on a domain, for example, they have different levels of granularity or different aggregation levels, finding the minimal ontology commitment becomes a difficult task [93] due to the number of heterogeneity conflicts that may arise [197]. Also, a single ontology approach is subject to changes in the data sources, which can affect the conceptualization of the domain represented in the ontology. Since it is not possible to build a common vocabulary that is general enough to cover all the different bioinformatics sources, and is also specific enough to offer translation support, such drawbacks have led to the use of multiple ontology approaches [197].

Although ontologies can resolve semantic heterogeneity problems, broaden the scope of searches that need to be carried out on integrated data sources, and enhance the quality and integrity of data to be integrated from heterogeneous sources, there are factors that limit their use. Firstly, ontologies can be incomplete in their representation of a domain due to incomplete ISA links, Part-Of hierarchies, incomplete lexicons, or missing concepts. Secondly, computational tools that compute a mapping between data in sources and ontology concepts are still immature and may not be easy to apply effectively [44]. Moreover, the lack of a common vocabulary makes it difficult to compare different source ontologies [197], which use different representations.

Furthermore, since the understanding of biological systems keeps changing, and the technical domains crossed by genomics and bioinformatics are disparate, there are always difficulties in capturing all the information in biological systems [194]. Thus, the different ontologies can become divergent in definition of terms [101].

Because different systems (for example, SEMEDA, TAMBIS, BACIIS) use different ontologies, there is a clash between them due to differences in terminology and other types of domain difference. Wiederhold [201, 202] describes four types of domain difference:

- *Terminology*: different names are used for the same concepts.
- *Scope*: similar categories may not match exactly; their extensions may intersect, but each may have instances that cannot be classified under any of the other.
- *Encoding*: the valid values for a property can be different, as different scales could be used.

- *Context*: a term in one domain can have a completely different meaning in another [101, 102].

In more recent work [153] identify ways to handle these mismatches.

3.3.2.3 Cross-referencing or Hard Links

Another integrating strategy for bioinformatics data sources is through the use of hard links. Hard links are used to link entries in disparate data sources. For example, if an MGI entry is about the sequencing of a specific gene, a hard link is established between the MGI entry and the corresponding nucleotide entry in GenBank [27], as this provides additional information about the gene .

In this approach, a user queries a data source and the processing follows hypertext links to related information in other data sources [27]. Data entries in different data sources can have relationships expressed as links, or predefined cross-references. Cross-references between related entries in heterogeneous sources are stored either in the form of index files as in SRS [74, 138], or hypertext links as in Entrez [145, 175]. These cross-references are used to achieve interoperability of heterogeneous bioinformatics data sources. They can be represented either by an entry in an ontology or by a global unique identifier (e.g. LSID). Such links or cross-references are determined in several ways, such as a computation of similarity between sequences using alignment tools such BLAST, or by mining the literature to discover linkage [140].

Bleiholder et al. [36] discuss how links are added to data entries in bioinformatics data sources, and identify the following reasons:

- Researchers add them when they discover a confident relationship between items.
- Data curators add them as a sign of a structural relationship between two data sources.
- Computational tools, for instance, BLAST, add them when a similarity is found between two data entries.

In some existing integration systems, joining information held in different data sources is based on cross-reference links. For example, one may want to find all DNA sequences in EMBL [49, 105, 118, 137] or GenBank [30-32, 85, 86], for a protein found in Swiss-Prot [37]. This query requires a hard link join using the accession numbers listed as cross-references in the Swiss-Prot source to the accession numbers in EMBL and GenBank.

However, cross-references, or hard links, have several drawbacks. They are subject to naming and value conflicts. For example, if a curator changes or deletes an entry that is related to an entry in another data source, the link fails [36, 140]. Moreover, these links are syntactically poor because they are present only at a high level of granularity, i.e., at the data entry level. Also, they are semantically weak, because they do not provide any explicit meaning, and a user only knows the data entries are related in some way [36].

3.4 Existing systems

Bioinformatics data source integration systems differ from each other in several dimensions. We will characterise existing systems in terms of the dimensions in Table 3.1.

From the start of this research, the author kept a list of the bioinformatics integration systems described in the literature. This list is not necessary complete but is comprehensive and contains 30 systems at this point in time (March 2008). The most common architecture used in these systems is based on data warehousing architecture (30% of the systems). While systems like SEMEDA, P/FMD and TSIMMIS use a mediation architecture, other systems like k2/Biokleisli, DiscoveryLink and ISYS use a federation architecture. Unlike DiscoveryLink, TAMBIS offers a global schema and data reconciliation. A full comparison is given in Appendix A.

dimension	**description**
Integration approach	this is whether the system uses a data warehousing, federation or mediation approach [189].
Data model	a model describes in an abstract way how data is held in database management systems. The commonly used models are Hierarchical, Network, Relational, Object Oriented, or Object-Relational [55-57].
Level of transparency	refers to the degree to which the user is shielded from the underlying structure and the need to choose the required source to answer a user's query [189].
Integration degree	this is either loose or tight. A system is tightly coupled if all the schemas of the integrated sources are mapped to one global schema, whereas a system is loosely coupled if there is no global schema [189].
Materialisation	the process of copying data from a primary database to a replicate database.
Data types	types of data the integration system handles.
Query operators	refers to the operators in a user query that the integration system can handle.
User model	type of users who will use the system.
Data Source interface	how to connect to a data source.
Global schema type	the common schema describing the data content of a data warehouse or federation that holds integrated data from a number of data sources.
Number of sources	number of data sources involved.
Resolving heterogeneity	this refers to whether the integration systems resolve the heterogeneity between the sources and level of this resolution [189].
Domain	the nature of the data sources involved in the integration - gene databases, DNA sequences, other domains.
Ontology	this refers to the extent an ontology is used to resolve heterogeneity between sources.
Query planning	how the query execution plan will access different, autonomous sources and put the results from diverse data sources together to form the complete result
Query caching	a mechanism that allows users to use effectively the results of prior queries to answer a new query.
Query adaptive	a query processing system is designed to be adaptive if it receives information from its environment and determines its behaviour according to that information in an iterative manner.
System platform	the platform in which the integration system runs.
Domain schema	the domain terminology and any other information that is needed.
User interface	how users interact with the integration system.
Query language	the language in which users of a system can interactively formulate queries and generate results. It is based on the contents of the data sources.
API	is there an application program interface to the integration system.
Output format	the format of the output produced.

Table 3.1: dimensions used in characterising existing system

We present here a sample of existing bioinformatics integration systems described in the literature. It includes SRS [74, 138], DAVID [63], TAMBIS [24, 138, 182], and myGrid [183]. SRS is a link-based integration system, while TAMBIS is an Object Oriented multiDB query system and DAVID is a data warehousing system. These samples are chosen to show specialised solutions through to more general solutions. These were chosen as they are popular and are representative of integration systems that use different approaches, namely, Data warehousing, federation, Mediation and Link-navigation. myGrid and BioMOBY were chosen as being representatives of the state of the art. For each of these systems, an overview and discussion of their strong and weak points is provided.

3.4.1 SRS

The SRS - Sequential Retrieval System - [74] is a Bioscience product of LION. Initially, SRS was developed at EMBL and extended at the EBI. In 1999, it was acquired by LION Bioscience. Currently, it is one of the most widely used bioinformatics data source retrieval systems; it uses a link-driven approach. The system accesses different bioinformatics data sources and builds an index to integrate them. Each data source must be wrapped and indexed by Icarus, which is a special wrapper programming language within SRS [138]. It uses Icarus-based meta-data to describe each source [138]. Whilst SRS provides the user with some transparency regarding the location, connection protocols and query language of each source, it does not shield its user from the formats and conventions of the integrated sources. SRS has various strengths:

Extensibility: since it uses a flat file based indexing mechanism, adding new sources is easy and straightforward [74, 138, 173].

Flexibility: it has an easy-to-use graphical user interface that acts as a unified front end to access multiple data sources [74, 138, 173].

On the other hand, SRS has several weaknesses: firstly, it is a keyword-based retrieval system, rather than an information integration system, so it does not provide any transformation or further operability above the query user result; thus, the user has to use other tools (such as BLAST, FASTA) for further analysis. Secondly, it works only with flat files, XML and relational databases and it does not integrate other types of sources, such as Object-Oriented. Thirdly, it does not enhance the data semantically nor does it create a global schema over the data [74, 138, 173].

3.4.2 DAVID

DAVID is an acronym for Database for Annotation, Visualisation, and Integrated Discovery [63]. DAVID is an integration system comprising bioinformatics tools and data sources developed by the Laboratory of Immunopathogenesis and Bioinformatics at SAIC-Frederick, Inc. for the National Institute of Allergy and Infectious Diseases of the National Institute of Health in Bethesda in the USA. DAVID aims to integrate information-rich data sources to provide users with a functional annotation and analysis of large lists of genes including human, mouse, rat or fly genomes. It also integrates different mining tools with the system to assist users to discover the biological meaning of the gene lists that result from the analysis of microarray data or other high throughput genomic data. It provides excellent graphical reports and summaries. The data sources integrated in DAVID include GenBank, UniGene [166], RefSeq [115, 166], LocusLink [166], KEGG [117], OMIM [151], and Gene Ontology [18]. Its warehouse is an ORACLE database designed to hold the functional annotation of genes. It uses LocusLink accession numbers to link to the primary sources of annotation, which have further gene specific information. With DAVID, it is the responsibility of users to extract and identify the gene identifiers manually from the experimental datasets and feed them to the system.

DAVID has strengths: firstly, since it is based on data warehousing, its main advantage is that system performance tends to be much better than other approaches. Secondly, its access to heterogeneous data sources is not limited by communication and bandwidth factors. Thirdly, its reliability tends to be better than that of other systems because there are no dependencies on network connectivity or the availability of the underlying data sources.

On the other hand, it has certain weaknesses. Firstly, it is not usually possible to submit an ad hoc query. Secondly, since it uses a warehousing approach, it suffers data warehousing approach problems, such as the large maintenance effort, limited flexibility to accommodate changing requirements, which are expensive to implement, and it does not scale well to a large number of data sources. In addition, adding new sources may lead to a redesign and repopulation of the data. Moreover, DAVID does not provide biologists with up-to-date data as it depends on when the warehouse is updated. Since DAVID uses hard links as cross-references between sources, there is a problem when sources change their references. Thus, it uses an inflexible hyperlink navigation, which does not allow the user to choose a desired link between sources.

3.4.3 TAMBIS

TAMBIS [90] is an acronym for Transparent Access to Multiple Bioinformatics Information Sources. It is an integration system that is built on top of BioKleisli [58, 59] and uses an extensive ontology expressed in the description logic GRAIL – GALEN Representation and Integration Language [147, 169]. However, unlike BioKleisli, it resolves semantic heterogeneities. This system was the first to use an ontology to support the integration of bioinformatics data sources [138]. It allows biologists to formulate complex queries over multiple bioinformatics data sources using a common query interface. In TAMBIS, data source-specific CPL (Collection Programming Language) [208] queries are mapped onto a global schema that is an ontology. This ontology is also used for query construction and validation. Whilst the wrapper extracts the user's query results from remote sources, the mediator integrates the results and sends them to the user. TAMBIS provides a graphical user interface to formulate queries by browsing concepts over its domain ontology [155]. TAMBIS has more than 300 CPL functions defined by BioKleisli. Each CPL extracts only one type of data from a single remote data source; however, the approach fails when the access interface of a data source changes [28].

The main components of the TAMBIS architecture are [23, 182]:

- The biological concept model,
- The knowledge-driven graphical user interface,
- The source model,
- The query transformation module, and
- The query execution module.

The steps in processing user queries are as follows:

- User expresses a query in GRAIL, which is a declarative source-independent description logic.
- The GRAIL Query is translated into its GRAIL Internal Form (GIF).
- The GIF query is transformed into a source-dependent query in CPL, which is processed against the data sources.

TAMBIS's strengths are that it supports the transparency of remote sources and hides the sources from users, and that the domain ontology allows a user to formulate a query without having any knowledge of the underlying data source [29].

However, its weaknesses are that, firstly, it is not robust to changes in a data source since its main component, the mapping model, is implemented manually. Secondly, adding new sources into the system is not a straightforward process. Thirdly, its interface is complicated and requires the user to have TAMBIS expertise. Fourthly, CPL is hardwired into the system, which makes it difficult to use this query language from an external system. Fifthly, there is no API interface. Sixthly, TAMBIS supports only one input format, namely, a Java Applet [77].

3.4.4 myGrid

myGrid is a general solution that gives access to remote and disparate biological systems. It was started in late 2001 in Manchester, England [183]. It had been noted that biologists were spending time building applications when what they really wanted to do was to investigate the biology. myGrid was an attempt to facilitate access to computational tools, experiments and data sources for these researchers. It has a Web service-oriented architecture, and allows web access to various services, utilising its middleware suite of tools for conducting *in silico* experiments. A user interacts with myGrid through a toolkit containing components for managing bioinformatics experiments, which can be saved. Using a registry built on RDF and OWL ontologies, myGrid converts investigations into their resource components. An abstraction layer called Grid Services then handles the communication with each of these resources to obtain the required information. This hides from the biologist details of how each component works.

A weakness of myGrid is that it does not have a simple interface; its users have to interact with its toolkit, making myGrid difficult to use and preventing biologists from accessing the functionality of the system [68].

3.4.5 BioMOBY

The BioMOBY is an open source research project initiated to provide more interoperability between biological data hosts and analytical services. It began at a retreat of representatives from the model organism database community in September 2001 [206]. It aims to provide an architecture for hosts to:

- Exchange common data representation formats.

- Establish a mechanism to represent meaning or context for machine-accessible data and services.
- Describe biological services in terms of their input and output
- Support the discovery and distribution of biological data and bioinformatics services through web services.

The BioMOBY interoperability system consists of the following primary components [206]:

- MOBY Object and Service hierarchies: An ontology describing the relationships between Objects and Services.
- MOBY Objects: An ontology describing biological data structures.
- MOBY Service: An ontology describing bioinformatics services.
- MOBY Central: A Web Service registry that acts as a search engine which allows biologists to discover resources capable of executing the task they wish to undertake.

There are several workflow tools that can search and browse the BioMOBY registry, for example Taverna [109, 124, 158, 190] workbench and Gbrowse Moby [203].

Although BioMOBY allows greater interoperability between data sources [207], there are some limitations, for example, Service discovery is insufficient to describe all aspects of the web services that it supports [207]. It does not handle the problem of service providers changing their interfaces without updating the MOBY registry[204]. Cross-references are semantically poor to some extent and are treated equally under the current API. Moreover, BioMOBY lacks a flexible query tool that allows rich queries to be executed on the federated data as it does not support the Boolean operators (AND , OR and NOT) in queries [205].

3.4.6 Semantic Web for Life Sciences (SWLS)

The mission of the Semantic Web for Life Sciences community is to improve the ability to conduct hypothesis-driven experiments [91] and other bioinformatics analysis by utilization of web-accessible data sources and analytical tools. This is achieved by use of the Semantic Web technologies for life science [156, 163]. The W3C-led Semantic Web initiative has established several of the standards and technologies needed to achieve SWLS [91]. These include:

- The Life Science Identifier system (LSID): this was designed to provide a unique global identifier for entities. An LSID is independent, stable, persistent, and resolvable [48].
- The Resource Description Framework (RDF): it is a method for knowledge representation which provides flexibility and extensibility of resources description. RDF describes knowledge by decomposing it into small parts called triples, namely subject, object and predicate [168] . It can be represented as a graph using:
    - a node for the subject.
    - a node for the object.
    - an edge for the predicate, directed from the subject node to the object node.
- The Web Ontology Language (OWL): it provides a language to specify and define the type of objects and their relations with each other within ontologies [160].

A lot of work and research has been done in this project and several tools have been created. However, at the beginning of this PhD project the tools were immature and suffered from drawbacks so we could not use them in the PhD project. For example there was:

- a lack of semantic information about the relationships,
- no a standard RDF(S) data access mechanism,
- the cost of storing and querying RDF triples was high, and
- the adoption of LSID was in its infancy and is still not universal [91, 171, 213]. Thus this work was not available for use in this project.

SLM in its current form is able to use LSIDs as identifiers if the sources being analysed use LSIDs. With respect to RDF, SLM can use this resource framework, provided alterations are made to the code. SLM can take advantage of the SWLS's semantic information about a resource if it is available. This may require some changes in SLM.

3.5 Challenges

The integration challenge is that the existing approaches and strategies suffer from the following difficulties:

i) Linkage types are fixed and difficult to change as they are determined by wrappers in a data warehouse, the middleware component in a federation, or the code that executes the warehouse in mediation.

ii) Breaking of links: when a URL changes, the direct links to it have to be changed. This primarily affects Link-driven and navigation systems, but can occur in data warehousing and federation based systems.

iii) Changes in database contents in the source may not occur in the data used until a later time, so the results may not reflect the latest data. This affects data warehousing, but not the other approaches.

iv) Difficulties in linking to non-bioinformatics data sources: link-driven and navigation systems can handle this in a limited manner, while the other types of systems have problems.

v) Inability to support multiple types of relationship: all four approaches are subject to this limitation to some extent.

vi) Data sources frequently cannot be joined using simple term-matching or comparison operators. Even more sophisticated approaches, which use ontologies to enumerate joinable terms, are often not sufficient [94]. It is a better to find methods for flexible linkage that allow users to drive the integration process and change the linkages, as this allows the easy investigation of alternatives theories.

It is often the case that a user needs to be able to change linkages and experiment with them in different ways as part of an investigation to see what yields interesting results. Thus, it is important that it is easy for a bioinformatician to be able to change the linkage type flexibly, and adjust the linkage so they can try investigating different linkages to see which one if any matches the purposes of their research. A join should be undertaken to reflect a semantic relationship between objects, as semantic relationships between properties of concepts may solve data integration problems in the bioinformatics domain. This means that there is a need for a researcher to be able to create different types of linkages between bioinformatics sources easily so that it is easy to investigate the effect of different relationships.

3.6 Summary

In this chapter, general approaches to integrating heterogeneous bioinformatics data sources were discussed. These approaches were classified into two main categories: architecture and matching strategies. Each architecture was described briefly in section 3.3.1. The strategies used to link data across data sources were discussed in section 3.3.2. Several bioinformatics data source integration systems that have been reported in the literature were then critiqued to identify why a more flexible framework is needed in this area of research. In the next chapter (Chapter 4), we introduce our proposed approach - the Soft Link Model (SLM).

Chapter 4

Soft Link Model

4.1 Synopsis

To address the challenges identified in section 3.5, it is proposed that semantic relationships based on the properties of concepts may solve many of the data integration problems in the bioinformatics domain. This chapter starts by introducing comparative genomics, its importance as a domain, and various types of biological relationships. The proposed Soft Link Model (SLM) approach is then introduced, in which integration is based on relationships between concepts, not just on field-values. A feature of the SLM approach is that the user can customize the linkage of data sources, by creating his/her own Soft Link Model, which reflects a linkage to be investigated in the research.

4.2 Comparative genomics

Comparative genomics is the study of relationships between genomes of different species and the analysis and comparison of these genomes [100]. It is usually undertaken to discover new properties of genes.

Comparative genomics offers opportunities to draw on information from historically distinct disciplines, to link disparate biological kingdoms, and so bridge basic and applied science. Cross-species comparisons are increasing the understanding of how genes are structured, and how gene structure relates to gene function, and how changes in DNA have contributed to the planet's biological diversity

[150]. This has led to new computational methods being developed that investigate chromosomal organisation, structure and homology.

By integrating functional and sequence data across species, biologists are able to annotate the genome of a species by using functional data from other species. Furthermore, comparative genomics provides evidence of close evolutionary relationships between gene families. According to Adjaye and his collaborators,

> *The advantages of cross-species comparison are two-fold. First, cross-species gene-expression comparison is a powerful tool for the discovery of evolutionarily conserved mechanisms and pathways of expression control. The advantage of cDNA microarrays in this context is that broad areas of homology are compared and hybridization probes are sufficiently large so that small inter-species differences in nucleotide sequences would not affect the analytical results. This comparative genomics approach allows a common set of genes within a specific developmental, metabolic, or disease-related gene pathway to be evaluated in experimental models of human diseases. Second, the use of microarrays in studies of mammalian species other than human and rodents may advance our understanding of human health and disease* [6].

Currently, 40 to 60% of the genes found in new genomic sequences do not have assigned functions. Some functions can be deduced by computational-structure determination and protein folding, but many research problems remain to be solved in this area [107]. Thus, computational methods will continue to play a major role in the functional annotation of genomes in the foreseeable future.

4.3 Biological relationships

Palakal and his collaborators define object and relationships as follows:

> *The term "object" refers to any biological entity such as a protein, gene, cell cycle, etc. and "relationship" refers to any dynamic action one object has on another, e.g. protein inhibiting another protein or one object belonging to another object such as, the cells composing an organ* [161].

A biological relationship can take several forms. In [52], the following classes of relationship are given:

- Evolutionary (for example, homolog, ortholog or paralog),
- Functional Genomic (for example, a biological process, a cellular component, or a molecular function),
- Structural,
- Phylogenetic,
- Mapping Terminology (Markers, Linkage, or Synteny),
- Genetic or Molecular Concept (for example, Genes, Polymorphisms),
- Containment, and
- Nomenclature (for example, gene A in species X = gene B in species Y).

We are concentrating in this thesis on Evolutionary Relationships (homolog, ortholog, paralog) and some of the Functional Genomic relationships (biological process, cellular component, molecular function), as they are used to discover remote evolutionary and functional similarities between gene products. Since evolutionarily-related genes are highly likely to share common aspects of function, a measurement of these relationships, which determines how similar they are, can be useful for gene functional annotation.

4.3.1 Homologous sequences

Homology is defined by Hillis as "*similarity due to inheritance from a common ancestor*" [104]. Two sequences are homologous if they share a common evolutionary history, i.e., there existed an ancestral molecule in the past that was ancestral to both of the sequences. A homolog can be either within the same organism (a paralog), or among different species (an ortholog) (see Figure 4.1).

4.3.1.1 Types of Homology

There are many types of homology [104], for instance:

- Orthology

  Orthologous genes are homologs that evolved as a result of a speciation event [104]. In other words, orthology is a homology that reflects the descent of a species [164]. Orthologous genes may or may not have the same function.

- Paralogy

  This is a homology reflecting the descent of genes. Paralogous genes are homologs that diverged as a result of a gene duplication event [104]. Paralogy may be distinguished from orthology by checking whether or not two homologs are found in the same individual [164].

- Xenology

  Xenologous genes are homologs that diverged as a result of a lateral gene transfer [104]. Antibiotic resistant genes are a classic example of Xenologs.

- Synology

  Synologs are genes that end up in an organism through a fusion of lineages [104].

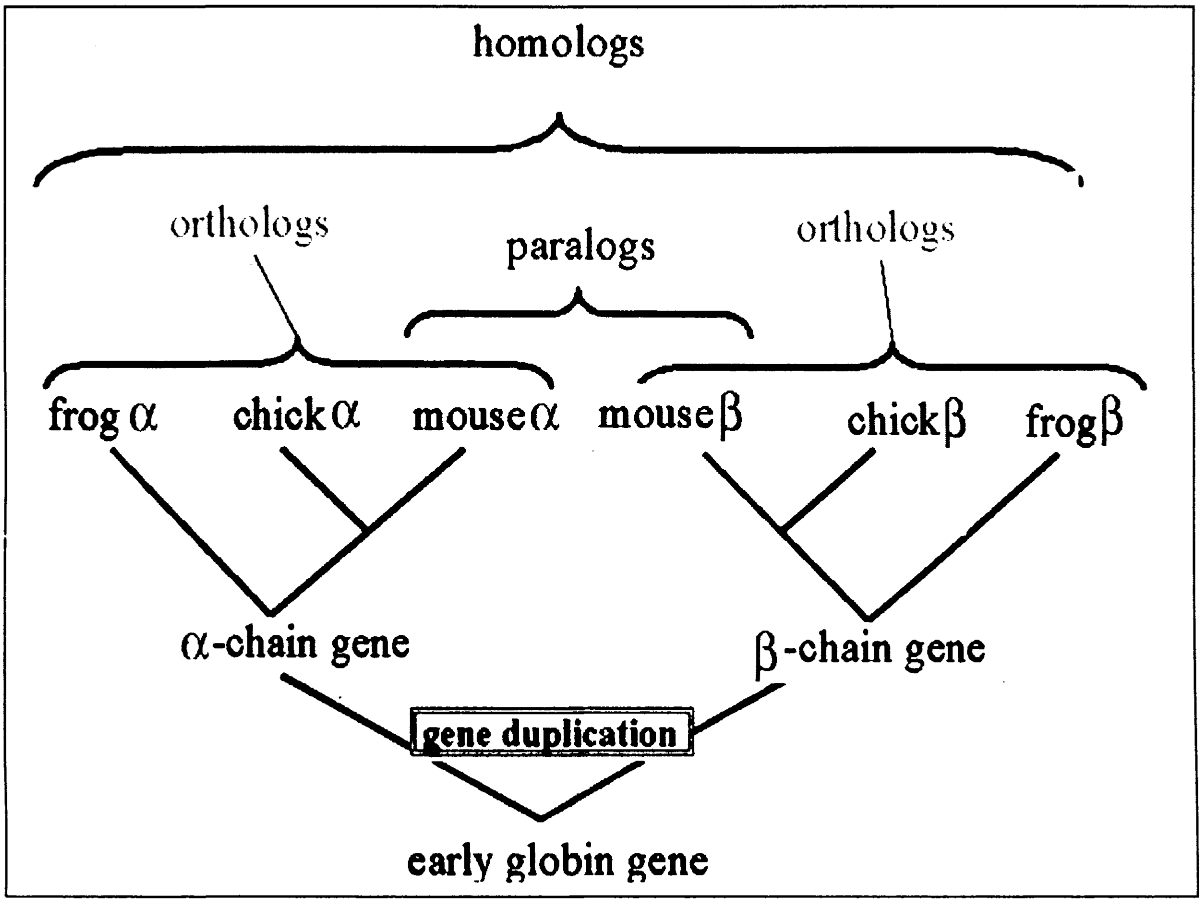


Figure 4.1: Orthologs and paralogs explained graphically [76]

4.3.2 Significance of the types of relationship

In this section, we present the relationship types used in SLM, and present also their place in the biological domain.

4.3.2.1 Homology

The search for homologous genes within organisms or across species is undertaken to identify genes that are similar. If a pair of genes is detected as homologous, and the properties of one are known, and the other has unknown properties, then the researcher can investigate whether the second gene has the same properties (i.e., functions, mechanisms and structure) as the first gene. Investigating the structures and functions of genes and proteins common to multiple species is an important focus of comparative genomics research [125], as it allows prediction of the functions of a new gene.

4.3.2.2 Orthology and Paralogy

Ortholog and paralog relationships are important for the following reasons:

- Ortholog relationships are important in determining functional equivalence [111].
- Paralog relationships can be used for function prediction. Paralogous genes are often involved in the same process, but have different molecular functions, for example, globins.

Thus, the results of orthology and paralogy support functional predictions and gene clustering. However, due to the complexity of biological problems and the lack of complete experimental and analytical models, there is a need to design automated knowledge-driven techniques to assist in the explanation and validation of predictive outcomes [198]. This is a driver of bioinformatics research.

4.3.2.3 GO-Based comparison

The automated comparison of complete sets of genes encoded in two genomes can provide insight into the genetic basis of differences in biological traits between species. The Gene Ontology (GO) consortium has created a common vocabulary to explain the relationships of gene products across species and to annotate genes for comparison purposes [178]. The inclusion of GO annotations in gene expression studies may explain why genes in a particular group share similar expression patterns, and it may help in identifying functionally-enriched clusters of genes [198]. The GO comprises three main ontologies:

Molecular Function (MF): The functions of a gene product are the jobs it does [89], for instance, binding. A pair of genes can have the same function if annotated by an equivalent GO term.

Biological Process (BP): This refers to the processes concerning living organisms [89], for instance, aging. A pair of genes can have the same biological process if annotated by an equivalent GO term.

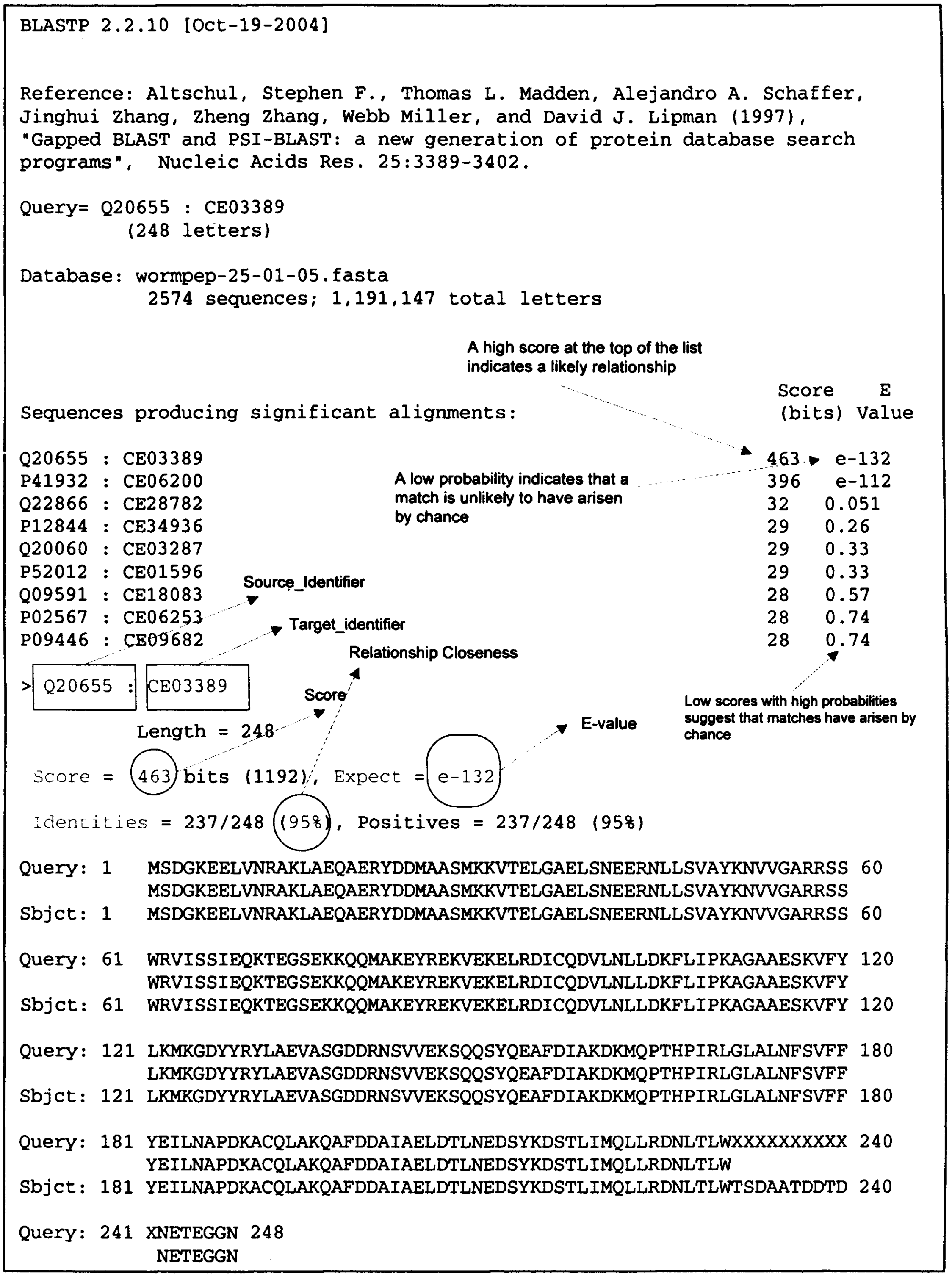
Cellular Component (CC): This describes locations, at the levels of subcellular structures and macromolecular complexes [89], for instance, cell. A pair of genes can have the same cellular component if annotated by an equivalent GO term.

4.3.3 Calculation of relationship closeness

The techniques used to measure the relationship closeness between a pair of concepts are presented in this section.

4.3.3.1 Homology closeness

The homolog relationship similarity closeness is expressed as the percentage of amino acid sequence identity between the protein sequences of a pair of gene products and is calculated using the BLAST algorithm. Similarity can be assessed by counting the positions that are identical between two sequences. As can be seen in Figure 4.2, significant information can be extracted from the BLAST output for each sequence pair. This information includes sequence identifiers, the score, the e-value and the identity between the two sequences. A high score at the top of the list indicates a likely relationship. Whilst a low probability indicates that a match is unlikely to have arisen by chance, low scores with high probabilities suggest that matches have arisen by chance.



```
BLASTP 2.2.10 [Oct-19-2004]


Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs", Nucleic Acids Res. 25:3389-3402.

Query= Q20655 : CE03389
         (248 letters)

Database: wormpep-25-01-05.fasta
           2574 sequences; 1,191,147 total letters


                                                                 Score    E
Sequences producing significant alignments:                      (bits) Value

Q20655 : CE03389                                                   463   e-132
P41932 : CE06200                                                   396   e-112
Q22866 : CE28782                                                   32    0.051
P12844 : CE34936                                                   29    0.26
Q20060 : CE03287                                                   29    0.33
P52012 : CE01596                                                   29    0.33
Q09591 : CE18083                                                   28    0.57
P02567 : CE06253                                                   28    0.74
P09446 : CE09682                                                   28    0.74

>Q20655 : CE03389
          Length = 248

 Score = 463 bits (1192), Expect = e-132
 Identities = 237/248 (95%), Positives = 237/248 (95%)

Query: 1   MSDGKEELVNRAKLAEQAERYDDMAASMKKVTELGAELSNEERNLLSVAYKNVVGARRSS 60
            MSDGKEELVNRAKLAEQAERYDDMAASMKKVTELGAELSNEERNLLSVAYKNVVGARRSS
Sbjct: 1   MSDGKEELVNRAKLAEQAERYDDMAASMKKVTELGAELSNEERNLLSVAYKNVVGARRSS 60

Query: 61  WRVISSIEQKTEGSEKKQQMAKEYREKVEKELRDICQDVLNLLDKFLIPKAGAAESKVFY 120
            WRVISSIEQKTEGSEKKQQMAKEYREKVEKELRDICQDVLNLLDKFLIPKAGAAESKVFY
Sbjct: 61  WRVISSIEQKTEGSEKKQQMAKEYREKVEKELRDICQDVLNLLDKFLIPKAGAAESKVFY 120

Query: 121 LKMKGDYYRYLAEVASGDDRNSVVEKSQQSYQEAFDIAKDKMQPTHPIRLGLALNFSVFF 180
            LKMKGDYYRYLAEVASGDDRNSVVEKSQQSYQEAFDIAKDKMQPTHPIRLGLALNFSVFF
Sbjct: 121 LKMKGDYYRYLAEVASGDDRNSVVEKSQQSYQEAFDIAKDKMQPTHPIRLGLALNFSVFF 180

Query: 181 YEILNAPDKACQLAKQAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWXXXXXXXXXX 240
            YEILNAPDKACQLAKQAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLW
Sbjct: 181 YEILNAPDKACQLAKQAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWTSDAATDDTD 240

Query: 241 XNETEGGN 248
              NETEGGN
```

A high score at the top of the list indicates a likely relationship
A low probability indicates that a match is unlikely to have arisen by chance
Source_Identifier
Target_identifier
Relationship Closeness
Score
E-value
Low scores with high probabilities suggest that matches have arisen by chance

Figure 4.2: A sample part of a BLAST output showing the pair of sequence identifiers, score, e-value and identities between each pair of the sequences. The identity's percentage can be used as the measure of relationship closeness.

4.3.3.2 Orthology closeness

The ortholog relationship closeness is expressed as the percentage of amino acid sequence identity between the protein sequences of a pair of gene products in different species and is also calculated using BLAST.

According to Huynen and Bork,

> *"orthologs then are defined in the following manner: (i) they have the highest level of pair wise identity when compared with the identities of either gene to all other genes in the other's genome; (ii) the pair wise identity is significant (E, the expected fraction of false positive, is smaller than 0.01), and (iii) the similarity extends to at least 60% of one of the genes"[112].*

4.3.3.3 Paralogy closeness

The paralog relationship closeness is expressed as the percentage of amino acid sequence identity between the protein sequences of a pair of gene products in the same species and is also calculated using BLAST.

4.3.3.4 GO-Based closeness

To estimate the semantic similarity between two genes g_i and g_j annotated with sets of GO terms A_i and A_j respectively, we calculate initially the similarity between the two GO terms. In the following section, we present different approaches for measuring the GO terms similarity.

4.3.3.4.1 Traditional edge-counting

An edge-counting approach calculates the distance between the nodes associated with the GO terms in a hierarchy: the shorter the distance between the terms, the higher the similarity. An example of this approach is Wu and Palmer's method [212], which uses the formula:

$$sim(t_i,t_j) = \frac{2N}{N_i + N_j + 2N} \qquad (1)$$

where N_i and N_j are the number of edges between t_i and t_j and their closest common parent in the GO hierarchy, T_{ij}, and N is the number of links from T_{ij} to the GO hierarchy root.

This similarity measure can be transformed into a distance by:

$$d(t_i,t_j)=1-sim(t_i,t_j) \qquad (2)$$

This is used to calculate the average inter-set similarity between each pair of t_i and t_j using:

$$d(g_k,g_m)=\underset{i,j}{avg}(d(t_{ki},t_{mj})) \qquad (3)$$

The GO-based similarity between two gene products g_k and g_p, is defined as:

$$d(g_k,g_p)=\underset{i,j}{avg}(\frac{2N}{N_{ki}+N_{mj}+2N}) \qquad (4)$$

The edge-counting approach is theoretically fairly simple. However, there are limitations, as it relies heavily on the idea that nodes and links in the GO are uniformly distributed. Although the approach is intuitive and direct, it is not sensitive to the depth of the nodes for which a distance is being calculated.

4.3.3.4.2 Information-theoretic

This measures the similarity between terms, based on the Information Content (IC) associated with or shared by the terms. The information content of a term is a value obtained by estimating the probabilities of occurrence of this term in a large corpus [116]. Thus, the more information two terms share, the more similar they are. Several techniques are based on this principle and these are summarised here.

Resnik:

This measure, created by Resnik[170], uses only the IC of the shared parents.

$$sim(t_1,t_2) = -\ln(\min_{t \in S(t_1,t_2)} \{p(t)\}) \qquad (5)$$

Where $S(t_1,t_2)$ is the set of parent terms shared by t_1 and t_2; and $p(t)$ is the probability of occurrence of t or its children in the database. The measure varies in value between infinity (for very similar concepts) to 0.

Lin:

Lin's technique [142] uses the IC of the shared parent and the IC of the query terms.

$$sim(t_1,t_2) = \frac{2x[\ln(\min_{t \in S(t_1,t_2)} \{p(t)\})]}{\ln P(t_1) + \ln P(t_2)} \qquad (6)$$

Where $S(t_1,t_2)$ is the set of parent terms shared by t_1 and t_2, and $p(t)$ is the probability of finding t or any of its parents in the database [19]. This measure generates a normalized value between 0 and 1.

Jiang:

The Jiang method [116] uses the IC of the shared parent and the IC of the query terms.

$$sim(t_1,t_2) = 2\ln(\min_{t \in S(t_1,t_2)} \{p(t)\}) - (\ln P(t_1) + \ln P(t_2)) \qquad (7)$$

Where $S(t_1,t_2)$ is the set of parent terms shared by t_1, and t_2,and $p(t)$ is the probability of finding t or any of its parents in the database [19]. This measure generates a semantic distance that can vary between infinity and zero.

Equations 5, 6 and 7 rely on the IC values assigned to the concepts in a hierarchy, but there are minor differences in the definitions. Lin and

Jiang use the IC of the shared parent and that of query terms whereas Resnik uses only the IC of the shared parent.

Once similarity between terms is measured, as above, the gene similarity is calculated by aggregating the similarity values obtained from the annotation terms of the genes [144] . Given a pair of gene products, g_k and g_p, and sets of annotations A_k and A_p consisting of *m* and *n* terms respectively, the between-gene similarity, *SIM(*g_k*,* g_p*)*, may be defined as the average inter-set similarity between terms from A_i and A_j

$$SIM(g_k, g_p) = \frac{1}{m \times n} \times \sum_{t_i \in A_k, t_j \in A_p} sim(t_i, t_j) \quad (8)$$

Where sim(t_i,t_j) is the similarity between the terms, which can be calculated using Equations 5, 6 or 7 [144] .

In our work, we use the average term-term similarity measure [144] because we are interested in the overall similarity between a pair of proteins rather than between pairs of ontology terms. Hence, in our work, the semantic similarity measure created by Lin [142] is used to determine the relatedness of each gene pair because it generates a normalized value between 0 and 1. However, all three techniques are implemented in our system, which allows bioinformatician to choose an appropriate technique. Given a pair of gene products, G_i, G_j which are annotated by a set of molecular function terms, T_i, T_j respectively, where T_i and T_j consist of m and n terms respectively, the relationship closeness between the genes is calculated using Equation 9 so the relationship closeness becomes:

$$RC(G_i, G_j) = \frac{1}{m \times n} \times \sum_{t_k \in A_i, t_p \in A_j} sim(t_k, t_p) \quad (9)$$

Three measures of relationship closeness, each based on information individually extracted from each of the GO hierarchies, namely, Biological Process (BP), Molecular Function (MF) and Cellular Component (CC), are implemented in the SLM. The related-biological-process relationship and the related-cellular-component relationship are calculated in the same way as is the related-molecular-function relationship

4.4 Soft Link Model

In this section, we introduce the Soft Link Model (SLM), our novel way of addressing the challenges. We start by defining the relationships, concepts and types of linkage implemented in the prototype of the SLM. These cover the most commonly used linkages in comparative genomic research. New types of linkage can be added to the prototype in the future. By following the SLM approach, we are able to increase the flexibility of linkage, reduce the time needed for the analysis of several experimental datasets, and eliminate some of the manual tasks.

4.4.1 Definitions

Definition 1: C = $\{c_1, c_2, \ldots\ldots c_n\}$ is a set of concepts, where a concept, c_i, represents a class of things in a real-world. Examples of concept are gene, protein, or species. Each concept has instances. The instance is an entry in a database that represents a real-world entity. Examples of instances are *Aeyo* gene (age of eyelid opening) or Cage1gene (cancer antigen 1) in the Mouse Genome Database.

Definition 2: **Relationship Closeness** (RC) measures the closeness of two instances of concepts, where 'closeness' is defined in terms of different dimensions. It measures the degree of closeness, i.e., how two instances of concepts are related to each other. It is expressed as a percentage, with 100% meaning c_1 is the same as c_2. A high value of RC indicates there is a significant link between the instances of concepts, and a low value of RC indicates no link or no significant link between the instances of concepts.

Definition 3: $P = \{p_1,p_2,......p_n\}$ is a set of concept properties, where a property is an attribute of a concept, such as the sequence string of a specific gene or the name of a specific protein. A property $p_i \in P$ is a unary relation of the form p_i (c_i), where $c_i \in C$ is a concept associated with property p_i.

Definition 4: R is a set of semantic relationships between the properties of concepts. Several types of relationship r can belong to R. Six types of relationship are implemented in SLM (see Table 4.1). New types of relationships can be added to the prototype in the future.

homolog
ortholog
paralog
molecular function
biological process
cellular component

Table 4.1: types of relationship supported in SLM

Definition 5: $G = \{g_1, g_2,......g_n\}$ is a set of algorithms. These algorithms include BLAST, similarity matches and other mining tools, and are used to calculate the strength of a type relationship between instances of concepts. These algorithms look at all possible pairs of specified concepts from the data sources and assign a relationship closeness score to each pair. If this score is above a cut-off or threshold value, then the relationship is accepted. This value can be adjusted in an iterative investigation to increase or decrease the number of matches and can be set to appropriate values for an investigation.

4.4.2 Formal Representation

The Soft Link Model (SLM) consists of concepts, instances, relationships and degrees of linkage. The SLM models the linkage between data sources in terms of concepts, properties and semantic relationships, and is formally defined as: SLM = (c_i, c_j, R, RC) where c_i, c_j are concepts, R is a type of relationship, and RC is the relationship closeness for the linkage between the instances of the concepts. The relationship between two instances of concepts is determined by considering the different properties (p_i, p_j) of the concepts. It can be formed by the syntax: R = ($p_i(c_1)$, $p_j(c_2)$, g, t) where $p_i(c_1)$ is a property (for instance, sequence, name) of an instance of the first concept, $p_j(c_2)$ is a property of an instance of the second concept, g is an algorithm used to calculate the relationship, and t is a cut-off score or threshold value.

SLM can be modelled as a graph G = (V, E), where V is a set of nodes and E is a set of edges (Figure 4.3). Concepts are represented by nodes, and relationship types between concepts are represented by edges between nodes. Relationship edges indicate that each instance of a concept (for example Mouse genome) may have a relationship with instances of the connected concept (for example C. elegans genome), and vice versa. Homology is a bidirectional relationship. For example if gene A from Genome B is homologous to gene C from Genome D then gene A is homologous to gene C. However, the encoded-by relationship is a unidirectional relationship. For example, If Protein P encoded by Gene A, it is not true that Gene A is encoded by Protein P. Relationship may be uni-directional or bidirectional. The relationship types in Table 4.1 are bidirectional. The closeness is represented by a label under the edge (Figure 4.3). The label of the node is given by a string, which represents a concept name. The label on the edge represents any user-defined relationship.

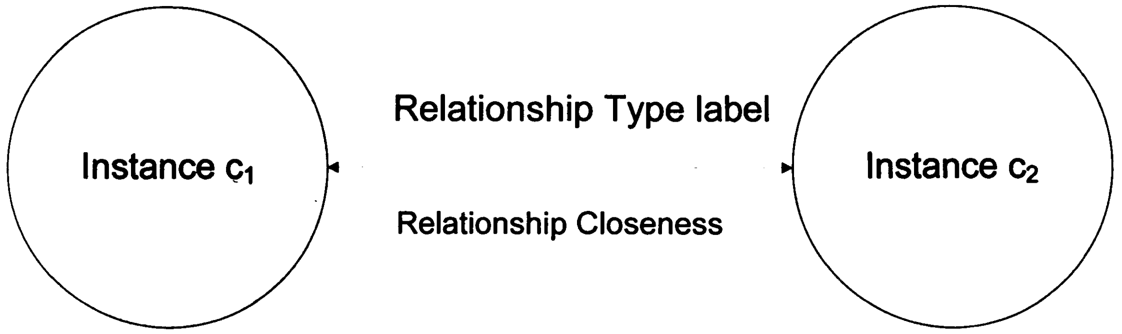


Figure 4.3: Representation of Soft Link Model. The symbol (◂▸) denotes that the relationship can be a uni-directional or bidirectional relationship where c_1 and c_2 are instances of concepts.

To mine for Evolutionary (homolog, ortholog or paralog) relationships between two genes, the sequence similarity can be applied. As discussed in sections 4.3.3.1, 4.3.3.2 and 4.3.3.3 the BLAST algorithm can be used to compute relationship closeness between two gene products using their sequences.

For example, if there are two data sources representing the gene annotation of different species: Mouse and C. elegans. Mouse's gene with two properties: *Accession and SQ*. C. elegans genes with two properties: *ID and Sequence* (Table 4.2 and Table 4.3). To mine for a possible homolog relationship between the different instances in these data sources, a BLAST algorithm will be used. The properties: *Sequence* and *SQ* will be used by the algorithm. Depending on the nature of the sequence (DNA or Amino Acid), different BLAST programs for the database search can be used. They are: blastn, blastp, blastx, tblastn and tblastx[ref]. The BLAST algorithm identifies homologous sequences by searching databases using the query sequence of interest. After the BLAST algorithm completes the search, the biologist will receive a report specifying found homologous sequences and their alignments to the query sequence. Figure 4.4 shows an excerpt of the blastp program report used to find possible homologue between Mouse gene *MGI:1891917* and C. elegans gene *WP:CE06200*. From this report, some useful information can be extracted:

- the identifier of the query sequence (*WP: CE06200*);
- the identifier of the database sequence (*MGI:1891917*);
- identities which represent the number and fraction of total residues which are identical. The identities percentage is used as the relationship closeness measure, and
- Expect value cutoff (-e) and Score are used as threshold values. A high score at the top of the list indicates a likely relationship (Figure 4.4).

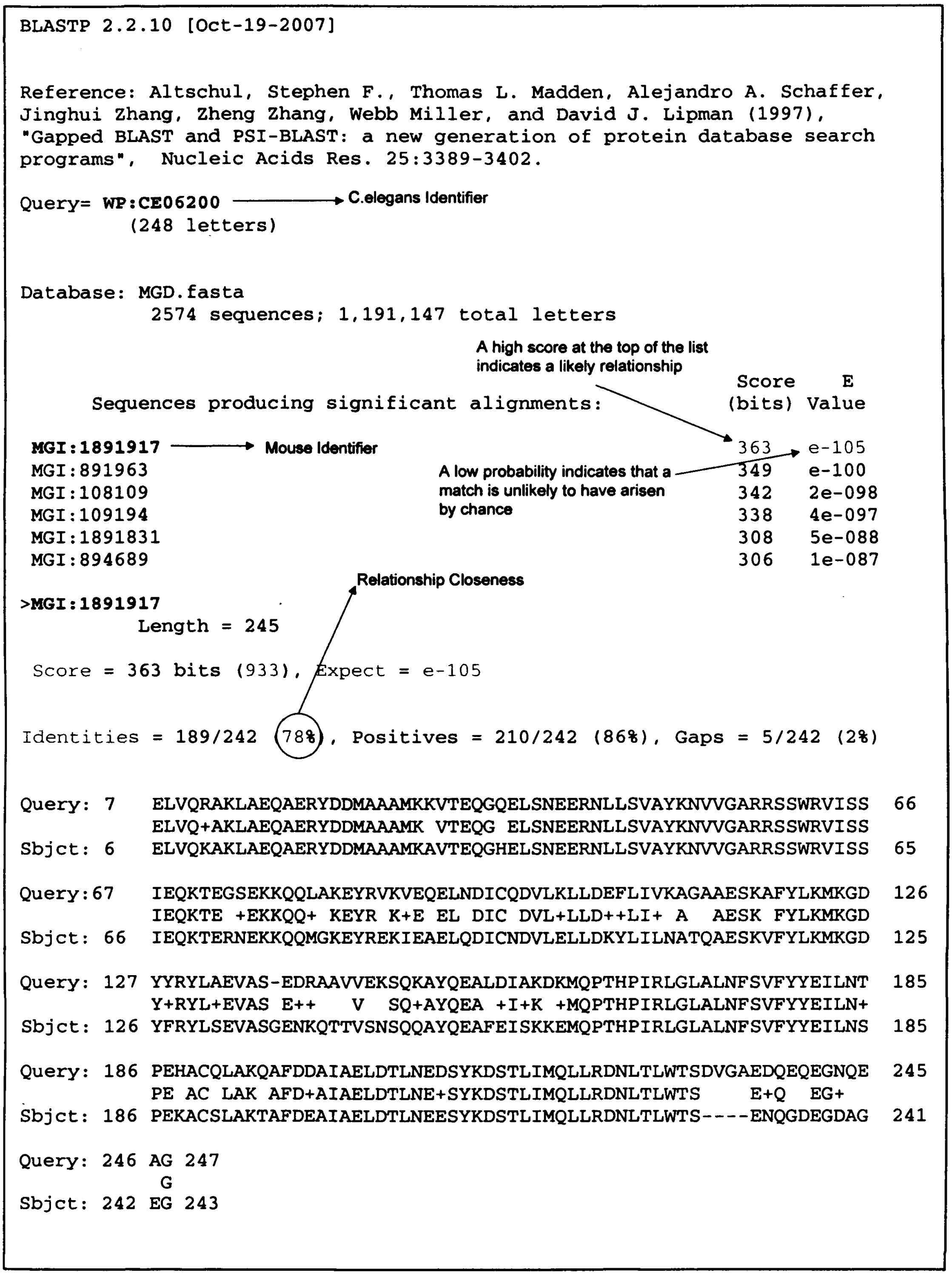
As can be seen from the report, the relationship closeness between Mouse gene *MGI:1891917* and C. elegans *WP:CE06200* is 78%. So homologs, orthologs and paralogs between genome are detected using BLAST similarity search.

ID	SEQUENCE
WP:CE06200	ELVQRAKLAEQAERYDDMAAAMKKVTEQGQELS........
WPCE24473	MCLVNEFVSN SNMKPALNVS GDEKELILQL...........
......	....

Table 4.2: Sample of gene annotation of C. elegans genes

Accession	SQ
MGI:1891917	ELVQKAKLAEQAERYDDMAAAMKAVTEQGHELS.......
MGI:891963	ELVQRAKLAEQAERYDDMAAAMKKVTEQGQELSN.......
....	.....

Table 4.3: Sample of gene annotation of Mouse genes



```
BLASTP 2.2.10 [Oct-19-2007]


Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

Query= WP:CE06200
          (248 letters)


Database: MGD.fasta
           2574 sequences; 1,191,147 total letters

                                                                   Score    E
      Sequences producing significant alignments:                 (bits) Value

 MGI:1891917                                                        363   e-105
 MGI:891963                                                         349   e-100
 MGI:108109                                                         342   2e-098
 MGI:109194                                                         338   4e-097
 MGI:1891831                                                        308   5e-088
 MGI:894689                                                         306   1e-087

>MGI:1891917
           Length = 245

 Score = 363 bits (933), Expect = e-105

Identities = 189/242 (78%), Positives = 210/242 (86%), Gaps = 5/242 (2%)

Query: 7    ELVQRAKLAEQAERYDDMAAAMKKVTEQGQELSNEERNLLSVAYKNVVGARRSSWRVISS   66
            ELVQ+AKLAEQAERYDDMAAAMK VTEQG ELSNEERNLLSVAYKNVVGARRSSWRVISS
Sbjct: 6    ELVQKAKLAEQAERYDDMAAAMKAVTEQGHELSNEERNLLSVAYKNVVGARRSSWRVISS   65

Query:67   IEQKTEGSEKKQQLAKEYRVKVEQELNDICQDVLKLLDEFLIVKAGAAESKAFYLKMKGD   126
            IEQKTE +EKKQQ+ KEYR K+E EL DIC DVL+LLD++LI+ A  AESK FYLKMKGD
Sbjct: 66   IEQKTERNEKKQQMGKEYREKIEAELQDICNDVLELLDKYLILNATQAESKVFYLKMKGD   125

Query: 127 YYRYLAEVAS-EDRAAVVEKSQKAYQEALDIAKDKMQPTHPIRLGLALNFSVFYYEILNT   185
            Y+RYL+EVAS E++   V  SQ+AYQEA +I+K +MQPTHPIRLGLALNFSVFYYEILN+
Sbjct: 126 YFRYLSEVASGENKQTTVSNSQQAYQEAFEISKKEMQPTHPIRLGLALNFSVFYYEILNS   185

Query: 186 PEHACQLAKQAFDDAIAELDTLNEDSYKDSTLIMQLLRDNLTLWTSDVGAEDQEQEGNQE   245
            PE AC LAK AFD+AIAELDTLNE+SYKDSTLIMQLLRDNLTLWTS     E+Q  EG+
Sbjct: 186 PEKACSLAKTAFDEAIAELDTLNEESYKDSTLIMQLLRDNLTLWTS----ENQGDEGDAG   241

Query: 246 AG 247
             G
Sbjct: 242 EG 243
```

C.elegans Identifier
A high score at the top of the list indicates a likely relationship
Mouse Identifier
A low probability indicates that a match is unlikely to have arisen by chance
Relationship Closeness

Figure 4.4: an excerpt of the blastp program report used to find possible homologue between mouse sequences and C.elegans sequences.

To mine for Functional Genomic (MF, BP, and CC) relationships between two genes, the semantic similarity measure can be applied. In our study, we will compute similarity between pairs of gene products rather than between pairs of GO terms. As discussed in section 4.3.3.4, Resnik, Jiang, and Lin's measures can be used to compute Semantic Similarity between two gene products. Each gene product may be annotated by a number of GO terms (Table 4.4 and Table 4.5). For example, if we had two genes: *MGI:99674* and *WP:CE38270* respectively, annotated by different Molecular Function GO terms (*GO:0000287, GO:0004016, GO:0004383*) and (*GO:0000166, GO:0004143, GO:0000166, GO:0004143, GO:0005515, GO:0008270*) respectively, and Lin's measures is used. The Lin similarity will be 83.8171023365594 (Table 4.6). This similarity represents the relationship closeness between Mouse gene *MGI:99674* and *WP:CE38270* in SLM model.

ID	Molecular Function
WP:CE38270	GO:0000166, GO:0004143, GO:0000166, GO:0004143, GO:0005515, GO:0008270
WP:CE38130	GO:0003700, GO:0003677
……	….

Table 4.4: Sample of gene annotation of C. elegans

Accession	MF
MGI:99674	GO:0000287, GO:0004016, GO:0004383
MGI:99676	GO:0003700, GO:0003677
…….	…..

Table 4.5: Sample of gene annotation of Mouse genes

C. Elegans Identifier	Mouse Identifier	Relationship Closeness
WP:CE38270	MGI:99674	83.8171023365594
WP:CE38270	MGI:99676	86.53620688314562
WP:CE38130	MGI:99676	86.81354710951456
WP:CE38130	MGI:99674	84.46357563224241

Table 4.6: The result of applying Lin's measure to compute semantic similarity between pairs of gene products using Molecular Function GO terms annotation of genes in Table 4.4 and Table 4.5.

4.4.3 SLM Operators

This section provides a formal definition of the SLM operators.

4.4.3.1 DiscoverR

This is a binary operator used to discover relationships between two instances of a concept. It investigates whether there is a relationship between a pair of properties (attributes) of the objects based on a specified algorithm. If the relationship closeness or the similarities between them pass a threshold value, it will consider there is a relationship between the concepts and the degree of this relationship is determined by the value of the relationship closeness, computed by the algorithm that was used to find the relationship. Thus the relationship table:

$$\text{SLM} \leftarrow \text{<pair of properties>}\ \Re\ \text{<algorithm,T>}\ (\text{C1,C2}) \qquad (1)$$

where

SLM represents the relationship table,
C1 represents the first concept,
C2 represents the second concept,
ℜ represents Discover Relationship operator, *DiscoverR,*
properties are the properties (attributes) used to discover relationships ,

algorithm is algorithm used to calculate the relationship between the properties (attributes). It computes the relationship closeness between each pair of properties, and

T is a threshold.

The relationship data are stored in a table as SLM (1st identifier, 2nd identifier, RC).

4.4.3.1.1 Relationship Discovery

The relationship types and how to discover them are covered in this section.

For the sake of simplicity, we assume concepts C1 and C2 are represented as two different relation tables (R and S) of data with several attributes(r and s).

$$R\{r_1,r_2,\ldots,r_i,\ldots,r_n\}$$

$$S\{s_1,s_2,\ldots,s_i,\ldots,s_m\}$$

There are two general approaches based on using the sequence or GO attributes.

In a sequence-attribute approach, an algorithm, for example, BLAST, is applied to a sequence attribute in R (say r_i) and a sequence attribute in S (say s_j). This returns a set of values for each pair in the alignment of the sequences (r_i, s_j) in the Cartesian product of R and S, R*S. These values are

E the e-value of the Homolog (r_i, s_j),

I the identity of the Homolog (r_i, s_j), and

S the score of the Homolog (r_i, s_j).

These define several subsets of the Cartesian product of R and S (Table 4.).

homolog	The e-value of Homolog (R*S) is the subset of R*S satisfying $E(r_i , s_j) <$ threshold value. This is a homolog link.
homolog	The identity of Homolog (R*S) is the subset of R*S satisfying $I(r_i , s_j) >$ threshold value. This is an alternative homolog link.
homolog	The score of Homolog (R*S) is the subset of R*S satisfying $S(r_i , s_j) >$ threshold value. This is an alternative homolog link.
ortholog	An Ortholog is created by applying to the e-value set E(R*S) the identity operator, namely: The $\text{Ortholog}((r_i , s_j)) = I(E(r_i , s_j))$ and it is selected if $I(E(r_i , s_j)) >$ threshold value and the sequences are from different species.
Paralog	A Paralog is created by applying to the e-value set E(R*S) the identity operator. The $\text{Paralog}((r_i , s_j)) = I(E(r_i , s_j))$ and it is selected if $I(E(r_i , s_j)) >$ threshold value and the sequences are from the same species.

Table 4.7: Different subsets from the Cartesian product of R and S of each pair in the alignment of the sequences (r_i, s_j)

When using a GO approach, there is an attribute in each relation, r_g in R and s_g in S, that has a set of GO terms as its values. These terms can be Molecular Function, Biological Process or Cellular Components.

These lists are then compared using a comparison algorithm (*Ontology-driven similarity algorithm),* which is based on the GO structure and techniques described in 4.3.3.4.2, and which calculates the relationship closeness values for the *Ontology-driven similarity* (r_g, s_g). This returns a set of values for the (r_g, s_g) in the Cartesian product of R and S, R*S.

If the *Ontology-driven similarity* (r_g, s_g) score > threshold then an appropriate relationship (biological process, molecular function or cellular component) has been established.

4.4.3.2 SoftJoin

The softJoin is a binary operator that is used to link two concepts. It is based on the relationship type and relationship closeness value.

$$S \leftarrow (C1 \underset{(RT,RC>t)}{\otimes} C2) \qquad (2)$$

where

C1 is the first concept,
C2 is the second concept,
RT is relationship table,
RC is the relationship closeness,
t is a threshold, and
⊗ is the softjoin operator.

4.4.3.2.1 Integration

We assume there is a set of data sources, where a data source has various types of concepts. For the sake of simplicity, we assume the data sources' schemas are implemented in a relational model and each concept in a data source is a relational table, and we use the following notation:

> A schema of Relation R of degree n is denoted by $R(A_1,A_2,\ldots,A_n)$ where $A_1, A_2,\ldots, A_n$ is a list of R's attributes.
>
> An n-tuple t in a relation R is denoted by $t=<v_1,v_2,\ldots,v_n>$, where v_i is the value corresponding to attribute A_i in the n-tuple.
>
> $t.A_i$ refers to the value v_i in tuple t of attribute A_i
>
> $S=\{ s_1,s_2,\ldots,s_n \}$ is a set of data sources
>
> $C=\{ c_1,c_2,\ldots,c_m \}$ is a set of concepts(relations) within a data source.

Thus,

$(s_i, c_j) \in SxC$ where s_i is a data source and c_j is a relation representing a concept in the data source. To integrate experimental datasets with public bioinformatics sources to annotate genes using SLM the following steps are taken.

Step1: A user feeds the system with the following input - Experimental dataset, Relationship type, Display fields.

Step 2: The system links the experimental dataset with public data sources to annotate a gene list, so the output is gene annotations. To explain the integration process, we use the following notation:

L is a set of dataset entries (set of experimental data entries)

Within L, a group of dataset entries that satisfy some conditions will be selected; L_c is for instance a list of mouse genes, where the expression is upregulated in response to aging. The attribute's name and metadata will be extracted from this experimental dataset. We define a function *ExtractMetadataOfExperimentalDataSet* to extract the experimental datasets' metadata. The approach for extracting this metadata is described in Section 6.1

A_c is an attribute of the metadata of L_c

E is the *ExtractMetadataOfExperimentalDataSet* function

A_c=E(L_c) = {A1,A2,,,An}

The potential linkage key for the experimental dataset will be determined, which will be used to link the experimental dataset with the public data source to enrich the gene annotation. We define a function *getLinkageKey,* which determines the potential linkage key of the experimental dataset. The approach for determining the potential linkage key is described in Section 6.1.

L_k is the linkage key of the dataset (L_c), where $L_k \in A_c$

L_k = getLinkageKey(L_c,A_c)

D_a is a display of the attribute list (which the user wants to retrieve)

SLM: a relation table stores the relationships between the pair of concepts across the data sources. SLM has the following attributes: (s1,s2,c1,c2,RT) where

s1, s2 are the pair of sources, c1, c2 are a pair of concepts, RT is the relationship type.

S_p is the primary source.

C_k is the concept (relation table in the source)

The primary source is selected by an algorithm from registered data sources with the system based on the experimental data type and the relationship type. The source selection algorithm is described in section 4.5. It selects the source that has the maximum relationships with the others sources, having the user concept and relationship type.

Linkage keys are extracted from the experimental dataset L_c and stored in a relation table G. This is done by a projection operation on L_c.

$$G \leftarrow \pi_{L_k}(L_c) \qquad (3)$$

G is then joined with the related relation in the primary source to get result.

$$\text{Result} \leftarrow G * S_p.C_k \qquad (4)$$

A projection is then made according to the user preference displayed in the attributes list. The resulting relation is the *PrimaryDataSet.* It contains only the attributes specified in D_a (the display attribute list)

$$\text{PrimaryDataSet} \leftarrow \pi_{D_a}(\text{result}) \qquad (5)$$

Related sources and concepts, which have the specified relationship type with the primary concept and source, are then selected from SLM and stored in a new relation *SR*.

$$SR(S, C, RT) \leftarrow \sigma_{(S_1 = S_p \wedge C_1 = C_k \wedge RT = rt)}(SLM) \qquad (6)$$

The defined operation *SoftJoin* is then used to link the primary result data set (*PrimaryDataSet*) with related data in other data sources.

For each tuple $t_i \in$ SR, i=0,,,n (number of tuples in SR)

$$\text{RelatedDataSet}_i \leftarrow \pi_{Da}(\text{primaryDataSet} \underset{<ti.v3,RC>t)}{\otimes} (t_i.v_1).(t_i,v_2)) \qquad (7)$$

The final result is obtained by the union of the primary dataset with datasets generated from related sources.

$$\text{FinalDataSet} \leftarrow \text{PrimaryDataSet} \cup \text{RelatedDataSet}_i \qquad (8)$$

4.4.3.3 Other operators

SLM also has the following operations:

a) Add a relationship to SLM:

$\text{SLM}_{New} = \text{SLM}_{Old}\ \text{U}\ \{\text{R}\}.$

b) Remove a relationship to SLM:

$\text{SLM}_{New} = \text{SLM}_{Old} - \{\text{R}\}.$

c) Add an instance to RKB:

$\text{RKB}_{New} = \text{RKB}_{old}\ \text{U}\ \{\text{r}\}.$

d) Remove an instance from RKB:

$\text{RKB}_{New} = \text{RKB}_{old} - \{\text{r}\}$

A user can suggest a new relationship by providing the System Administrator with the following information: pair of data sources, pair of concepts, relationship type, relationship closeness, and pair of identifiers for the data sources.

4.5 Source selection algorithm

The system selects the sources, which answer the user query based on the parameters in the query: species, concept and relationships. In describing the algorithm, we assume the following:

$C= \{c_1,c_2,\ldots,c_n\}$ is a set of concepts

$S=\{s_1,s_2,\ldots,s_m\}$ is a set of sources

$(s_i,c_j) \subset S \times C$

$R = \{R_1,R_2,\ldots,R_k\}$ is a set of relationships. These relationships are either internal relationships (between concepts in the same source) or external relationships (between concepts in different sources).

C_q is a set of concepts used in the user query $C_q \subset C$

R_q is a set of relationships used in the user query $R_q \subset R$

The first step is to find the sources that have the user query concepts, and then find the concepts in those sources that have the user query relationship. Instances of these are retrieved having the RC defined in the user query. The algorithm is shown in Figure 4..

4.6 Summary

Since comparative genomic explanations provide a more comprehensive understanding of both the complex structures and diverse functions within the genomes of different organisms, this chapter presented an approach to the integration of data across species to assess genomic comparison based on similarity knowledge extracted from the GO-driven functional annotations and sequence similarities.

The approach is based on the calculation of relationship closeness values, which originate from each of the GO hierarchies and homology and its types. The advantage of this method lies in the application of prior biological knowledge to estimate the relationship closeness between genes. In addition to homology closeness, this chapter introduced three hierarchy-specific relationship closeness measures, each based on information individually extracted from each GO hierarchy (BP, MF and CC).

The next chapter describes the system architecture for the proposed Soft Link Model introduced in this chapter.

```
Algorithm: Source Selection
Input: query $q$
Output: Sources St
1: parse $q$; (get concepts $c_q$, relationships $r_q$)
2: upload SLM;
3: identify the number of data sources participating in the integration system;
4: for all data source $Si$ do begin
        No_of_Relationship =0;
        For each concept in $Si$ do
            if $c_l$ in $C_q$ then
            for each relationship r in $Cj$ do
                if relationship $r_j$ in $R_q$ then
                    No_of_Relationship ++;
                end if
            end for
        end for
      if No_of_Relationship > 0
            St (targeted sources) = St U {Si}
      end if
    end for
5: find S $\in$ St with maximum No_of_Relationship;
6: return S;
```

Figure 4.5: source selection algorithm

Chapter 5

System Architecture

5.1 Introduction

This chapter describes the system design for an "illustration-of-concept" system of the SLM model. In this chapter, we overview the overall **I**ntegration and **D**ata **M**ining of **B**ioinformatics **D**ata sources (IDMBD) system's architecture, describe its components, and explain how the components are connected. The architecture is based on the conceptual model and approach described in Chapter 4. The system architecture with its phases and components is described. Integration of data sources utilizing the SLM model is accomplished in two phases: phase 1 - relationship mining and discovery, and phase 2 – data source linkage and integration. Each phase and its components are explained in detail. The current prototype is built to work with two particular species, but can be easily extended to handle more species and to link to information in data sources such as disease and pharmaceutical. The steps the administrator/user has to follow to add linkage for such a data source to the IDMBD and so enhance the SLM system are detailed in section 5.3. The system's stages are the primary focus of attention in section 5.4, where the eighteen steps needed to answer a user query and link experimental datasets are summarised. The interaction between the mediator and SLM to enhance gene annotation and provide a user with relevant information from other related sources is detailed in section 5.5. The chapter concludes with a summary of the chapter.

5.2 System architecture

The components of the system architecture interact and work together to achieve its design aims. The system operation, the role of its components and the information exchanged among components are described here. The architecture is based on mediation as proposed by Wiederhold [201]. The pragmatic interest in creating this architecture is to reduce the amount of work required to introduce a new source by the creation of the corresponding wrapper [25]. Consequently, the mediator allows extendibility by the addition of new data sources to the integration system. The mediator architecture preserves data source autonomy and supports access to up-to-date data as the mediator uses a wrapper's that encapsulate the underlying structure of the data sources, so that wrappers' access to data sources is transparent to the mediators. This preserves a data source's autonomy and gives a biologist easier access to these sources, while enabling him/her to retrieve the most up-to-date biological data. Thus the linkage to a new source is achieved by creating a wrapper for it. This means that the source is unaffected by the linkage and the IDMBD requires a new wrapper to be written.
Figure 5.1 shows the functional architecture and its main components.

5.2.1 Architecture layers

The system consists of four layers as shown in Figure 5.1: Client Application, Mediation, Wrappers and Data Sources. A user interacts with the mediator in the top layer to access indirectly any data sources. The mediator can be viewed as a bridge between the user/application and data sources. It performs the processing that is common to data sources. However, source-specific transformations are done in individual wrappers.

1. Client Application layer: client/applications reside here and interact with the IDMBD framework. The client consists of a graphical user interface (GUI) and is responsible for the generation of user queries and uploading experimental datasets. It has several tools that process and analyse an experimental dataset.

2. Mediation layer: this provides a transparent view of multiple heterogeneous data sources and coherent views of data in the data sources by performing semantic reconciliation through the Common Data Model (CDM) data representations provided by the wrappers. It merges the results from sources and returns them to users. Generally, it is responsible for data transformation and integration, and communicates with the client application layer and wrapper layer. Further details on its components are provided in section 5.2.2.2.6.
3. The wrapper layer provides access to the data in the data sources using the data source's API, translates user queries into source-specific queries, extracts data, and maps the results from data sources into the common data model of the integration system. The wrappers conceal technical and data model heterogeneities; the way wrappers access data sources is transparent to the mediator and this preserves a data source's autonomy. New wrappers can be added.
4. Data sources: heterogeneous data sources reside here. They can be accessed through wrappers. Data sources may be structured or semi-structured.

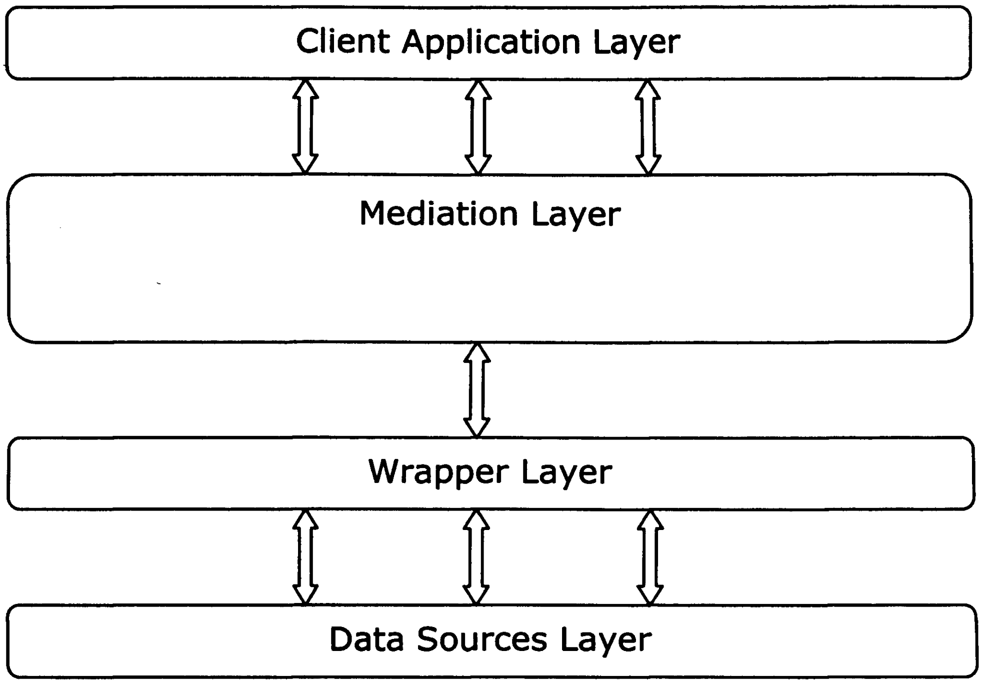


Figure 5.1: the IDMBD Framework: a conceptual view

5.2.2 Integration Phases

Integration of data sources using the SLM model is performed in two phases: Phase 1 - relationship discovery and data mining, and Phase 2 - data source linkage and integration.

5.2.2.1 Phase 1: Relationship Discovery and the SLM model

This phase discovers relationships in the data sources attached to the system. These data sources are varied. In this phase, relationships between biological objects are identified. Many tools are used to discover relationships, such as alignment tools, text matching or other data mining tools (for instance, classification and association rules). The first step is parsing the data sources to extract the attributes of interest, which are used to identify relationships. Having identified the concepts and attributes of a source-pair, the system will invoke the appropriate algorithm to discover any relationships that might exist between concepts. The algorithm being used or other data mining tools are responsible for calculating the degree of the relationships between a source-pair. Objects involved in the relationship are then stored in a knowledge base in triple form (Source id, Target id, Relationship Closeness). User preference (constraints, parameters, algorithms) are considered during the discovery of relationships and the building of the relationship knowledge base. The relationships' metadata will be stored in a relational table as source, target, relationship type, name of file containing actual data. The metadata describes the type of relationship, data sources and objects involved, and refers to the table storing the actual mapping. The user can adjust the parameters used to discover relationships and calculate the degree of the relationship.

This subsystem consists of the following components:

- Parser: parses data in the relationship discovery component.
- Relationship Discovery: mines and finds relationships between objects in different sources using appropriate algorithms, search tools, and data mining tools.
- Relationship Table Generator: creates a knowledge relationship base in a triple form (Source id, Target id, Relationship Closeness). Figure 5.2 shows the algorithm generating this knowledge.
- Search and Data Mining Tools: describes the features of the tools and algorithms that are used for relationship discovery, such as location, syntax, parameters, availability and other relevant metadata of input/output, resource requirements and constraints. It is also responsible for the choice of search mechanism and invoking data mining tools and algorithms. It stores an algorithm's metadata for use in relationship discovery.
- SLM builder: an SLM file consists of metadata describing the relationships between concepts. This component builds the SLM

metadata for a run and stores the relationship between each pair of concepts of the data sources, the relationship type and the actual relationship table name. Figure 5.3 shows the XML schema definition for the SLM with required and optional elements.

- Relationships Tables: These tables are the knowledge base storing the relationship instances between data sources in triple format (Source id, Target id, Relationship Closeness). They automatically generate database links.

```
Algorithm: Relationship Generator

Input: List E, S, cut-off  // E, S are a list of entries of a data source; cut-off is a constant
                              value

Output: RelationTable RT

1: for each e in E do begin

        for each  s in S do begin

              Do match (e,s)

              Score  =  score of match(e,s)

          If Score >= cut-off  then

                 RelationCloseness = Score

                 RT = RT U  {(e,s, RelationCloseness)}

          end if

        end for

     end for

2: return: RT;
```

Figure 5.2: Algorithm to generate a relationship knowledge base

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
  XML schema for Soft Link Model metadata.
 </xsd:documentation>
</xsd:annotation>
<xs:element name="SLM-knowledge-base">
<xs:attribute name="no" type="integer" use="required"/>
<xs:element name="database" minOccurs=0 maxOccurs="unbounded">
<xs:complexType>
<xs:element name="concept" minOccurs=0 maxOccurs="unbounded">
 <xs:complexType>
<xs:element name="relations" minOccurs=0 maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
 <xs:element name="SLM" minOccurs=1 maxOccurs="unbounded">
 <xs:attribute name="DBName" type="RC" use="required"/>
 <xs:attribute name="concept" type="String" use="required"/>
<xs:attribute name="RelationType" type=" relationships " use="required"/>
<xs:attribute name="File" type="String" use="required"/>
<xs:attribute name="FileType" type=" String " use="required"/>
</xs:sequence>
</xs:complexType>
</xs:complexType>
</xs:complexType>
<- ->
<xsd: simpleType name="relationships">
<xsd:restriction base="xs:string">
<xsd:enumeration value="homolog"/>
<xsd:enumeration value="ortholog"/>
<xsd:enumeration value="MolecularFunction"/>
<xsd:enumeration value="BiologicalProcess"/>
<xsd:enumeration value="CellularComponent"/>
</xsd:restriction">
</xsd: simpleType>

<xsd: simpleType name="filetype">
<xsd:restriction base="xs:string">
<xsd:enumeration value="mySQL"/>
<xsd:enumeration value="text"/>
<xsd:enumeration value="OO"/>
</xsd:restriction">
</xsd: simpleType>
</xs:schema>
```

Figure 5.3: XML schema for SLM metadata

5.2.2.2 Phase 2: Data source linkage and Integration

This phase enables a user to use the created relationships to integrate the data. Figure 5.4 overviews its architecture. There are seven modules.

5.2.2.2.1 User Interface

This provides a single access point for users to query data sources within the system. It allows a user to upload experimental datasets and enrich gene annotations. It hides the complexity of the underlying structure and data schema of data sources. Its goal is to enable the user to interact easily with the system. There is also a facility in the interface for the user to register new data sources with the system. This interface can accept a variety of different types of user query such as a gene identifier or a table of experiment results.

It allows the user to upload and integrate experimental datasets with the available data sources. The user can set his/her preference (relationship type, relationship closeness, and displayed fields). Once the data are uploaded, the user is prompted to choose the potential linkage key and required fields as well as the relationship type and relationship closeness. It facilitates also the creation of an SLM and the relationship knowledge base.

In brief, this module allows the user to:

- browse discovered relationships between entities/concepts across heterogeneous data sources using a tree-like display of relationships.
- create a distinct SLM to discover relationships between concepts and entities.
- upload experimental datasets and link them with available bioinformatics data sources to enrich gene annotation from different species.

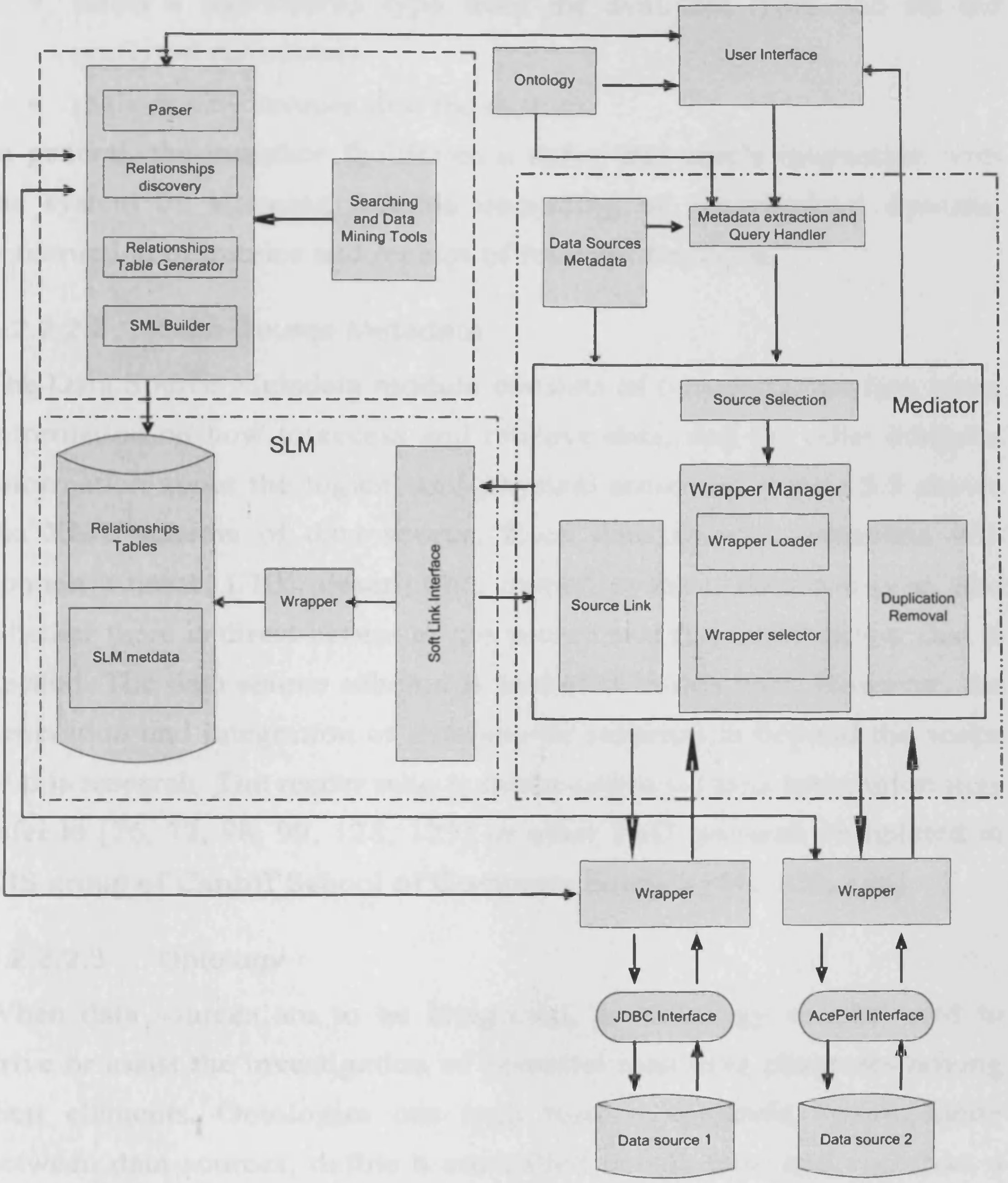


Figure 5.4: Overall Architecture of Integration system

- construct queries.
- select a relationship type from the available types and set the preferred parameters.
- register new sources into the system.

In general, the interface facilitates a naive end-user's interaction with the system by allowing flexible uploading of experimental datasets, construction of queries and receipt of relevant feedback.

5.2.2.2.2 Data Source Metadata

The Data Source Metadata module consists of two parts: the first gives information on how to access and retrieve data, and the other contains information about the logical and physical structure. Figure 5.5 shows the XML schema of data source. Each data source's metadata will contain a name, URL, description, owner, system, database type, and whether there is direct access to the source and the JDBC driver that is needed. The data source schema is included in this part. However, the generation and integration of data source schemas is beyond the scope of this research. The reader who is interested in schema integration may refer to [26, 72, 98, 99, 128, 129] or other PhD research completed in KIS group of Cardiff School of Computer Science [66, 122, 188].

5.2.2.2.3 Ontology

When data sources are to be integrated, an ontology can be used to drive or assist the investigation of potential matching processes among their elements. Ontologies can help resolve semantic heterogeneity between data sources, define a controlled vocabulary, and construct a query so that the user is unaware of the data source's structure.

We use a domain ontology in metadata extraction to enhance the metadata and find possible relationships between experimental datasets and domains (see Chapter 6).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
  XML schema for data sources.
 </xsd:documentation>
</xsd:annotation>
<xs:element name="database" type=" DB_info" minOccurs=1 maxOccurs="unbounded">

<xsd: complexType name=" DB_info">
<xs:sequence>
<xs:element name="ID" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="Name" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="Description" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="Owner" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="URL" type="xs:integer" minOccurs="0" maxOccurs="1"/>
<xs:element name="System" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="DataBase" type="xs:string" minOccurs="0" maxOccurs="1"/>
 <xs:element name="Direct_Access" type="xs:boolean" minOccurs="1"
maxOccurs="1"/>
<xs:element name="Host" type="xs:string"  minOccurs="1" maxOccurs="1"/>
<xs:element name="Port" type="xs:integer" minOccurs="1" maxOccurs="1"/>
<xs:element name="User Name" type="xs:string"minOccurs="0" maxOccurs="1" />
<xs:element name="Password" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="JDBC_DRIVER" type="xs:string" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xsd: complexType>
</xs:schema>
```

Figure 5.5: The XML Schema definition for data sources

5.2.2.2.4 Soft Link Module

The Soft Link Module mines and stores the relationships and cross-references between different objects. It provides a flexible linkage between data sources using the relationships created in phase 1.

The soft link module uses SLM metadata to find possible related sources. The module receives requests in XML format from the mediator. It then collects data from the SLM metadata and relationship tables in RKB and stores them in output XML. This XML becomes the response that is sent to the requesting mediator.

5.2.2.2.5 Metadata extraction and Query Handler

This component parses a user query and rewrites it in an appropriate format. It parses the experimental datasets, extracts metadata, and detects a suitable linkage key. It uses the domain ontology to enhance an experimental dataset's metadata (see Chapter 6 for more detail).

5.2.2.2.6 Mediator

The role of the mediator is to handle all communication to and from data sources. It also communicates with the soft link module to retrieve relationships between sources. It has four specific jobs:

- to try to find a suitable primary data source to satisfy a given query.
- to communicate with SLM, and query for possible related data sources.
- to invoke data source wrappers to send queries/deliver user queries to relevant data sources.
- to receive result sets, combine them and send the outcome to the user.

The mediator has the following modules:

- Source Selection This component selects an appropriate source to answer a user's query based on user preferences and query parameters. This component is responsible for the selection of suitable sources to answer user queries or annotate experimental datasets. It uses the algorithm in Figure 5.2 to select a suitable data source.
- Wrapper Manager: responsible for instantiating the wrappers the system is configured to use. It manages existing wrappers and performs many tasks: loading existing wrappers, communicating with the SLM to retrieve relationships between sources, and

removing duplicates during result assembly. It provides a basic service for managing a set of data source wrappers. It has two components:

- *Wrapper loader* – dynamically loads all available data source wrappers registered in the integration system and ensures they are loaded when the system starts.
- *Wrapper selector* – chooses and invokes wrappers for the selected data sources to answer a user query.

- Source links: The mediator interacts with the SLM to request other related sources for the primary result. This component handles communication between the mediator and the SLM. It sends an XML request to the SLM to fetch other related sources. Source links use a request/response paradigm to interact with the SLM.

- Duplicate removal: merges results, removes duplicates, and passes the combined result to the clients/users.

5.2.2.2.7 Wrappers

The wrappers provide access to remote data sources and transform the results into an integrated form. The wrappers conceal technical and data model heterogeneities. The method of access to wrapped data sources is transparent to mediators to preserve data source autonomy. The wrapper shields a user from the structure and complexity of data sources. There is one wrapper for each data source involved in the system, which provides access to data of a specific format. If a source allows direct access to its underlying RDBMS, a JDBC wrapper will forward an SQL query statement for processing by the source's database system. If the data source has a different interface, the wrapper will use an appropriate query format. For example, to access Wormbase, it will use the *AcePerl*, which is an object-oriented Perl interface for AceDB.

A wrapper performs many tasks, including:

- Using JCDB drivers or other standard APIs of the data source to connect the sources and to receive a result set.
- Submitting queries to the data source through SQL, native query language of the source, or as a series of source API calls.
- Providing a means to extract data from semi-structured sources (for example, flat files, HTML, text).

5.3 Building the SLM

The administrator/user builds an SLM by:

a) Identifying the concepts and properties to be used in the model.

b) Identifying appropriate relationship types between concepts and properties.

c) Setting the threshold for the relationship closeness measure.

d) Choosing algorithms to compute the soft link. An algorithm is required for the comparison of two concept properties. The variable used to measure the closeness of the biological entity's relationship should also be specified.

e) Creating RKB tables. This can be done in different ways, e.g., offline or on-the-fly at run time.

5.4 System Sequence

Figure 5.6 shows the steps taken by the system to answer a user query and annotate and link experimental datasets. The steps are described in Table 5.1.

Step (1): The mediator receives a user query/experimental dataset, and selects the primary source to answer the query.
Step (2): The mediator invokes the wrapper of the selected source.
Step (3): The selected source's wrapper connects to the data source by means of its API and submits the query to this data source.
Steps (4, 5): The source wrapper receives result sets from the data source and sends them to the mediator.
Step (6): The mediator extracts identifiers from the result sets then interacts with the Soft Link Model Adapter. It sends the source name, concept, identifier and user preference (the relationship which the user wants to use to link data sources and the relationship closeness threshold).
Step (7): The Soft Link module loads the SLM's metadata and determines, whether there are relationships associated with the concept and data sources sent to it by the wrapper manager.
Step (8): If a relationship specified by the user is found between the selected concepts of the data source and other concepts in other data sources, the SLM will pass the relationship table name to the mediator.
Steps (9, 10, and 11): The Soft Link module invokes the relationship wrapper, which opens a connection to the RKB and fetches instances satisfying user preferences. Basically, it fetches related concepts, data sources, and identifiers of the related entries in the related source.
Step (12): The Soft Link Adapter responds to the mediator with a list of related identifiers and source concepts.
Steps (13, 14): When the mediator receives the response, it invokes the wrapper of the related source and passes related identifiers to it.
Step (15): The wrapper connects to the related source by the data source standard API and submits a query.
Step (16): When the wrapper receives related data set results, it passes them to the mediator
Step (17): The mediator combines related dataset results with previous results and removes any duplicates. The mediator maps the data set results to the user view, i.e., when it receives results from individual sources; it integrates the results and sends them to the user.
Step (18): The mediator sends the datasets to the user.

Table 5.1: steps taken by the system to answer a user query

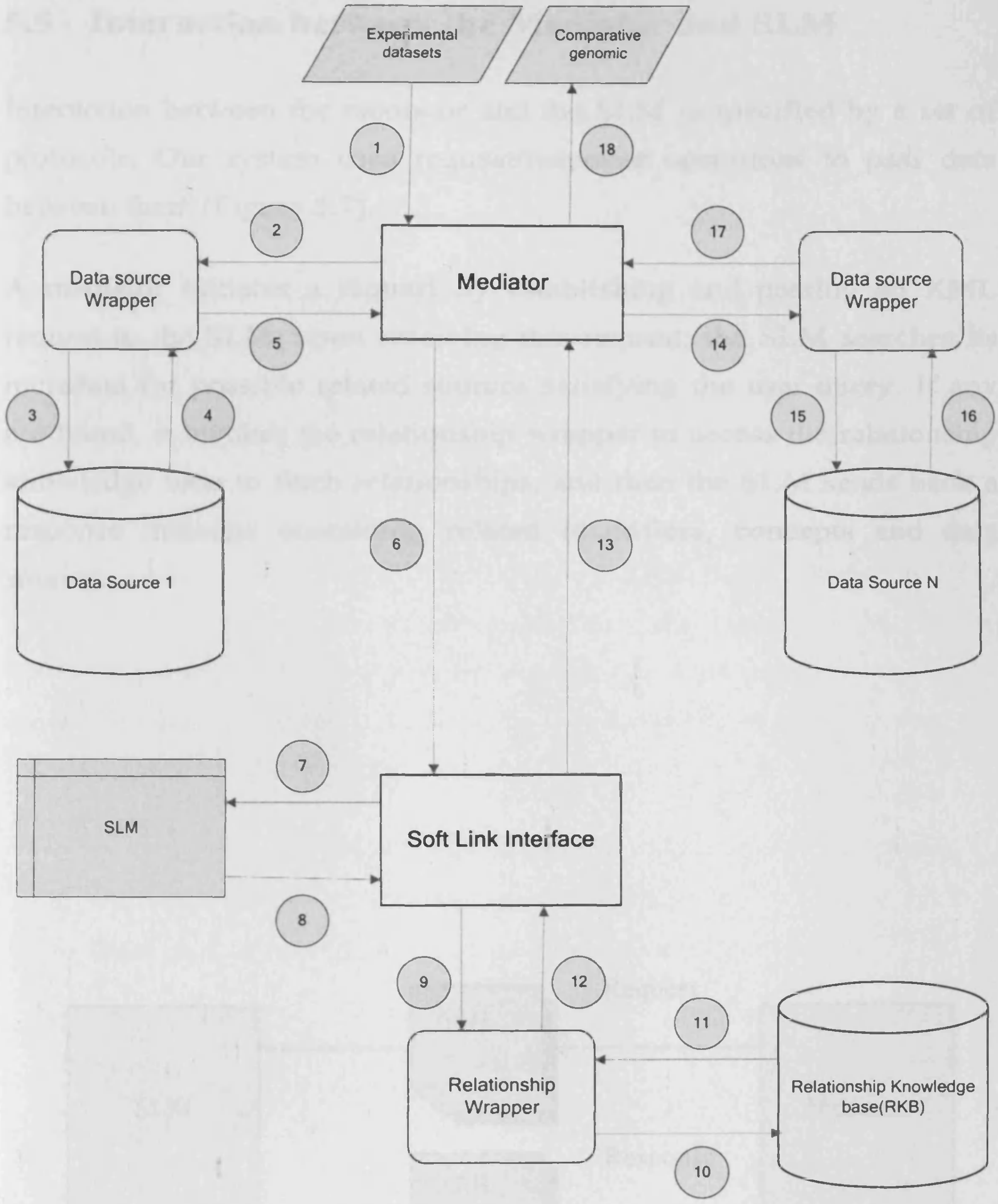


Figure 5.6: Sequence Diagram

5.5 Interaction between the Mediator and SLM

Interaction between the mediator and the SLM is specified by a set of protocols. Our system uses request/response operations to pass data between them (Figure 5.7).

A mediator initiates a request by establishing and passing an XML request to the SLM; upon receiving this request; the SLM searches its metadata for possible related sources satisfying the user query. If any are found, it invokes the relationship wrapper to access the relationship knowledge base to fetch relationships, and then the SLM sends back a response message containing related identifiers, concepts and data sources.

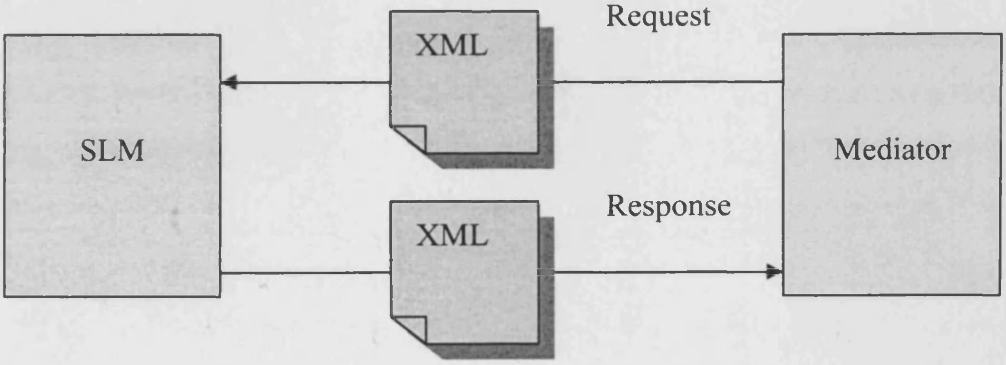


Figure 5.7: the mediator interacts with the SLM via a request/response paradigm

5.5.1 Request

The mediator sends a request to the SLM as an XML document. The request operation is used to find related data in other data sources. This XML document contains the following: the data source, the concepts, the relationship type, the relationship closeness, and a list of identifiers.

When the SLM receives requests in XML format from the Mediator, it collects data from the SLM metadata and knowledge relationship base and transforms them into XML. Figure 5.8 shows the XML schema definition for a Request operation.

5.5.2 Response

SLM responds to the mediator's request with an XML document. The XML is generated by the SLM after receiving and interpreting a request containing the related data source, concepts and identifiers. Figure 5.9 shows the XML schema definition for the Response operation with the following required elements:

- Identifiers: list of identifiers of related objects on other species,
- Concept: name of the related concept, and
- Data Source: name of the related data source.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xsd:annotation>
  <xsd:documentation xml:lang="en">
    XML schema for Request operation.
  </xsd:documentation>
</xsd:annotation>
<xs:element name="DataSource" type="xs:string"/>
<xs:element name="Concept" type="xs:string"/>
<xs:element name="relationship"/>
<xs:complexType>
<xs:attribute name="RelationshipType" type=" relationships" use=”required”/>
<xs:attribute name="RelationshipCloseness" type="RC" use="required"/>
</xs:complexType>
<xs:element name="identifier" minOccurs=1 maxOccurs="unbounded">
<- ->
<xsd: simpleType name="relationships">
<xsd:restriction base="xs:string">
<xsd:enumeration value="Homolog"/>
<xsd:enumeration value="Ortholog"/>
<xsd:enumeration value="Paralog"/>
<xsd:enumeration value="MolecularFunction"/>
<xsd:enumeration value="BiologicalProcess"/>
<xsd:enumeration value="CellularComponent"/>
</xsd:restriction">
</xsd: simpleType>

<- ->

<xsd: simpleType name="RC">
<xsd:restriction base="xs:decimal">
<xsd:minInclusive value="0"/>
<xsd:maxInclusive value="1"/>
</xs:schema>
```

Figure 5.8: XML schema definition for the Request operation

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsd:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xsd:annotation>
  <xsd:documentation xml:lang="en">
    XML schema for Response operation.
  </xsd:documentation>
</xsd:annotation>

<xsd:element name="PrimaryDataSource" type="xsd:DataSets" minOccurs=0
maxOccurs=”unbounded”/>
<xsd:element name=”comment” type “xsd:string”/>

<- ->
<xsd: complexType name="DataSets">
<xsd:sequence>
<xsd:element name="PrimaryDataSource" type="xsd:string"/>
<xsd:element name="PrimaryConcept" type="xsd:string"/>
<xsd:element name=" PrimaryIdentifier" type="xsd:string"/>
<xsd:element name=" RelatedIdentifier" type="xsd:RealtedData" minOccurs=”1”
maxOccurs=”unbounded”/>
<xsd:sequence>
</xsd:complexType>

<- ->
<xsd: simpleType name="relatedData">
<xsd:sequence>
<xsd:element name="RelatedDataSource" type="xsd:string"/>
<xsd:element name="RelatedConcept" type="xsd:string"/>
<xsd:element name="RelatedIdentifier" type="xsd:string"/>
<xsd:sequence>

</xsd: simpleType >
</xsd:schema>
```

Figure 5.9: XML schema definition for the Response operation

5.6 Summary

The IDMBD architecture, including its basic phases and components, was presented in this chapter. The two-phase integration of data sources utilised by the SLM model was explained, namely, the relationship discovery and data mining, and source linkage and integration. This architecture is based on the conceptual model and approach described in Chapter 4. The steps needed to answer user queries and annotate and link experimental datasets were described. Interaction between the mediator and SLM to enhance gene annotation and provide a user with the required information from other related sources was described in depth. An overview was given of how the system is built of components, and their connection

Chapter 6

Extracting Metadata of Experimental dataset

6.1 Synopsis

The process of automatically extracting metadata from an experimental dataset is an important stage in efficiently integrating this dataset with data available in public bioinformatics data sources. Metadata extracted from the experimental dataset can be stored in databases and used to verify data extracted from other experiments' datasets. Moreover, the biologist can keep track of the dataset so that it can be easily retrieved next time. This extracted metadata can be mined to discover useful knowledge; it can also be integrated with other information using a domain ontology to reveal hidden relationships. The experimental dataset may contain several kinds of metadata that can be used to add semantic value to linked data. This chapter describes an approach to extract metadata from an experimental dataset. It describes the metadata extraction phase (query handler component in Figure 5.4) of the IDMBD system [8-12], which we have developed to link experimental datasets with externally available data.

6.2 Introduction

Emerging technologies in biotechnology have made it possible to study thousands of genes or proteins in a single laboratory experiment [7, 181]. However, in order to find relevant biological knowledge from these experiments, it is important to analyse the experimental datasets as well as cross reference and link these large volumes of datasets with information available in external biological data sources accessible online to enrich gene annotations.

A significant challenge in this process is integrating gene annotation with gene expression and sequence information [136, 138, 193, 194]. Thus, biologists can study genes based on their function, chromosomal location, and tissue expression and also cross-reference this data with data from different species derived using diverse expression analysis platforms.

When linking and integrating data held in an experimental dataset in a semi-structured form with data held in external bioinformatics sources, it is essential to gather as much information about the experimental dataset as possible. This information can be found in the experimental dataset from column names and their contents as well as other types of metadata held in such a dataset.

6.3 Experimental dataset model

The system uses a three-phase approach: metadata extraction, schema creation and utilisation of a schema to link the experimental data with appropriate external data. In this section, we concentrate on the first of these phases and on how it is achieved.

6.3.1 Metadata extraction

Metadata are data about data that provide descriptive information about resources for the purpose of finding, managing, and using them more effectively [53, 149, 180]. Much of an experimental dataset is stored in an unstructured format, for instance, in a flat file with different data representations, either comma separated value (CSV) or tab delimited text, or some similar format. An experimental dataset may contain several types of metadata that can be utilised to add semantic significance to data linked with it. Examples of metadata are column names and row headers, which are usually specified in such files. To link these datasets with public bioinformatics sources, it is necessary to gather as much information about the datasets as possible. This can be achieved by making more use of metadata. Our approach makes use of the following types of metadata, which are located and extracted for the purpose of integrating the experimental dataset with available public bioinformatics sources.

6.3.1.1 Element name

Column headers are metadata indicating the main concepts that the file represents. Based on the data representation of the file and specified separator (tab, comma, space), the header line is converted into tokens. The number of tokens is then used to determine the number of elements to be extracted and the token value that contains the column header as the element name. The column heading is extracted from the experimental file to represent the element name.

6.3.1.2 Element structure type

Data structure type is detected by analysing the dataset vertically for each element in the dataset. Data structure types used are integer, string, date, and double. Each value in the element dataset will be checked to determine whether it is a string, an integer, a double or a date. An element is considered a string if at least one of its values consists of any character between a-z, A-Z, ' () +,-.?:/= and SPACE, for example, “bird”. The element is considered an integer if its values are a string of characters consisting of the digits 0-9, for example, number 20. The element is considered a double if its values can be converted to double format, i.e., contains a number and decimal point, for example, 56.15. The element is considered a date if its values can be converted to date format, for example, 01-01-2006. This is limited to a few example types.

6.3.1.3 Element length

By analysing the value vertically for each column and computing the maximum length of a representation in a column, the element’s length can be determined.

6.3.1.4 Constraints

It is necessary to identify the existing semantics of the data when possible. Constraints [57, 70] that may apply to the data need to be detected, for example, whether the "NOT NULL" constraint is specified for an element or not. Other constraints include whether the element's value is positive or negative, as in the case of integer or float values. In the present project, all element dataset values are scanned to check whether they could be null or not. An element is considered null if there is a complete absence of value within the column for at least one entry.

6.3.1.5 Candidate key for linkage

A candidate key is detected from an experimental dataset by analysing both extracted metadata and data values. Each candidate key has a certain set of characteristics that makes it suitable for the role of linkage key. These characteristics are 'not null', 'unique', 'single word', 'fixed length', and 'unambiguous'. Moreover, the name of an element that may be a linkage key should have a meaning and contain keywords such as *key, ID, number, No., accession, identifier.* The approach taken to detect the linkage key is determined by analysing the following:

- Element name: Usually, the creator of an experimental dataset file intends to use keywords to specify the candidate key in this data. In a biological experimental dataset, these keywords are "key", "number" as in (GenBank Accession Number), "No.", "identifier" as in (gene identifier), "accession" as in (Swiss-Prot Accession), "id" as in (Genbank ID, UniqSeqID, Clone Id), and other similar terms. Comparing a column name with this keyword list often gives an indication of the primary or candidate key in the dataset.

- Element value: In this step, the dataset is analysed vertically to capture the semantic significance and characteristics of each element. Five factors are taken into consideration:

i) *Uniqueness*: The linkage key must be unique within its domain [57, 70]. A key's main purpose is to help the user to identify one single entity in a data source, regardless of how many entities there are.

ii) *Not null:* Null is a known value and stands for "value is unknown" [57, 70]. The linkage key must always, without exception, hold a value that is NOT NULL.

iii) *Ambiguous:* The linkage key's value should be unambiguous. This value must not contain a value like "n/a" or "unknown" or "not available" or a special character like "?" or "-" or similar values.

iv) *Fixed length:* In biological sources, the primary key often looks very much the same in terms of format and length [93], for example, characters followed by numbers: P0496, DXS231.

v) *Brevity / Single word*: In most cases, primary keys are single words.

- Knowledge base: The column name can be used as a keyword to search for related semantic concepts in sources' metadata, integrated schema, and domain ontology. In this stage, an attempt is made to match elements of the experimental dataset with elements of the integrated schema and sources' schema. The column name is used to extract corresponding concepts from the data source schema and integrated schema. The column name is also used to extract corresponding concepts from the domain ontology and a search is made for the column name and all synonyms; for example, element "position" in an experimental dataset is a synonym for "location", and "species" may be equivalent to "organism" in the domain ontology. Another approach augments the column name with synonyms and searches the sources and integrated schema.

The relationship of each element in the experimental dataset with the integrated or sources' metadata should be specified to generate enhanced metadata as described in section 6.4.

We use a scoring system that assigns a score to each criterion. If more than one candidate is found, only the one with the highest score is considered. Table 6.1 shows the scoring system. A negative score is assigned to null. A difference is made among keywords that may be in the column's name. Key-words like "number", "accession", "key", and "identifier" have a high score since their existence in a column name suggests they are likely to be keys and suitable for linkage, whereas keywords like 'no' may occur by chance in the column name as part of a word, like "no" in "synonym" or "id" in "aid", "said", "solid", and "void".

Candidate keys will be ranked based on the criteria and semantic relations with the integrated or sources' metadata; for example, if the experimental dataset contains a Gen-bank accession number, gene identifier, and gene symbol, which is the most appropriate linkage key among the three for use as a link with the public bioinformatics sources to be used by the system in the integration process? The aim is to find the element that has the maximum score. This process can be represented mathematically as follows:

Let,

n: be the number of elements,

m: be the number of the criterion,

$S_{i,j}$: be the score of the j-th criterion of the i-th element,

T_i: sum score for the i-th element, and

MaximumScore is the maximum score across the element's total score.

Therefore:

$$T_i = \sum_{J} S_{i,j} \quad , \text{ for } j=1,2,\ldots m \qquad (1)$$

$$\text{MaximumScore} = \max T_i, \text{ for } i=1,2,\ldots n \qquad (2)$$

The **linkage key** is the element i which has the maximum T_i.

Criteria		**Score**
Unique		5
Null		-10
Column name	accession	15
	key	15
	identifier	15
	number	10
	id	8
	no	2
Ambiguous		-5
Single Value		5

Table 6.1: Scoring System

To reduce the effect of heterogeneity between different metadata elements and to improve integration, potentially similar elements that are detected must be converted to match each other in representation. A conversion function that converts the representation of detected metadata is used.

6.3.2 Schema creation

Once all metadata elements are extracted and all semantic relationships are detected, a schema for the dataset is constructed. This schema describes the data structure or type and some of the constraints, for example, element name, element type, element length, is it unique? is its value null? is it candidate key?

6.3.3 Schema exploitation

Once the schema of the experimental dataset has been constructed, the next step is to use this schema to generate a table in a relational form or as an XML document suitable for use, linkage and integration with other bioinformatics sources. The unstructured experimental dataset file is parsed so that it can be imported into a relational table or an XML document.

6.4 Metadata Linkages with Domain Ontology

This section discusses how to map metadata elements onto concepts of a domain ontology to enhance the metadata and discover any semantic relationships with the concepts in the domain ontology.

6.4.1 Ontology

When data sources are to be integrated, an ontology can be used in the potential matching processes among their elements [14, 93, 176]. It helps in discovering implicit and hidden knowledge through conceptualisation of a domain of interest, and in overcoming the effect of synonyms. Ontologies describe what the concepts are, and how they are related. They play an important role in supporting information exchange, reusing and sharing. In our work, a domain ontology will be used to facilitate the semantic integration of experimental datasets with public bioinformatics data sources and to make the data, especially metadata, machine readable, understandable and more easily linked according to the requirements of biologists.

An available domain ontology is the TAMBIS Ontology (TaO) [24]. It contains knowledge about bioinformatics and molecular biology concepts and their relationships. It does not include any instances. The stated aim behind designing this ontology is given as being

> *"to provide an ontology that could help underpin the development of systems that perform at least some of the functions of a domain expert. In general terms, these functions amount to knowing (i) what things are in the domain and (ii) when and how these things are related."*

6.4.2 Discovering semantic relationships

The relationships between concepts given in an ontology and an experimental dataset's metadata allow the flexible linkage of this dataset with heterogeneous data across distributed data sources. These relationships provide more flexibility in linkage by providing different links.

Once the candidate key is identified, a search is made using this candidate key or its synonyms from the domain ontology to find matching concepts/terms. We extract the concepts from the domain ontology to which the candidate key is related and all relationships associated with the concept. As the experimental dataset concepts are mapped onto related concepts in the domain ontology, we mine for relationships associated with each related concept in the domain ontology, as well as for concepts linked to concepts of the experimental datasets. For example, if the candidate key is AccessionNumber, we may find in the ontology domain relationships associated with this concept; for example, the following are linked to AccessionNumber:

isAccessionNumberOf, isIdentifierOf, isECNumberOf.

The algorithm for this process is shown in Figure 6.1.

Step 1: map the experimental dataset concept (candidate key) into concepts in a domain ontology. Many terms in the domain ontology may map into the candidate key.

Step 2: for each related concept in the domain ontology, mine for semantic relationships and associated concepts.

Step 3: associate the discovered semantic relationships and concepts with the experimental dataset concept (candidate key) to enhance metadata.

Figure 6.1: Algorithm for mapping experimental dataset elements to Ontology

6.4.3 Enhanced metadata

Experimental Dataset metadata will be enhanced with any semantic relationships discovered from the domain ontology. Enhanced metadata provides a flexible means for linking the experimental dataset with other public bioinformatics data sources. The enhanced metadata is represented as:

EnhancedMetadata=<C, SR>

where C represents the knowledge base concept derived from the domain ontology that is related to the candidate key and SR represents the semantic relationships that have been revealed.

For the sake of simplicity in this example, we assume the experimental dataset concept (candidate key) is mapped to only one concept in the domain ontology. Consider the domain ontology in Figure 6.2, the candidate key *AccessionNumber* is mapped to a similar concept *AccessionNumber* in the domain ontology (Figure 6.3 and 6.4). All relationships and concepts associated with the related domain ontology concept are extracted. So, the enhanced metadata for the candidate key is:

<AccessionNumber,{<gene,isAccessionNumberOf>,<protein,
isAccessionNumberOf>,<DNA,isAccessionNumberOf>}>

The biologists then determine which concept and relationship are of interest based on their experimentation and nature of the dataset. If there is ambiguity because there is more than one possible linkage, the system will display the alternatives to the user. The user selects the appropriate linkage; to help in this decision, the user is given additional information from the ontology.

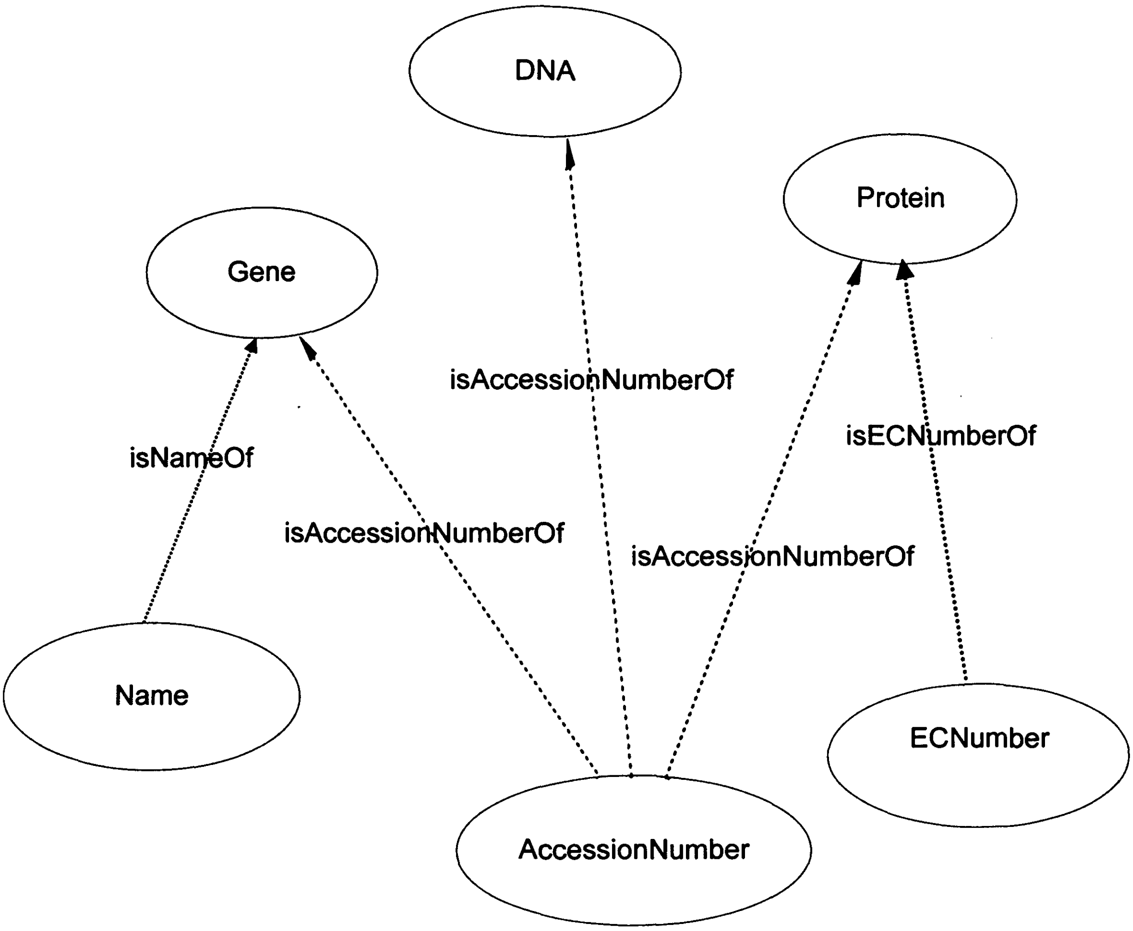


Figure 6.2: Domain Ontology

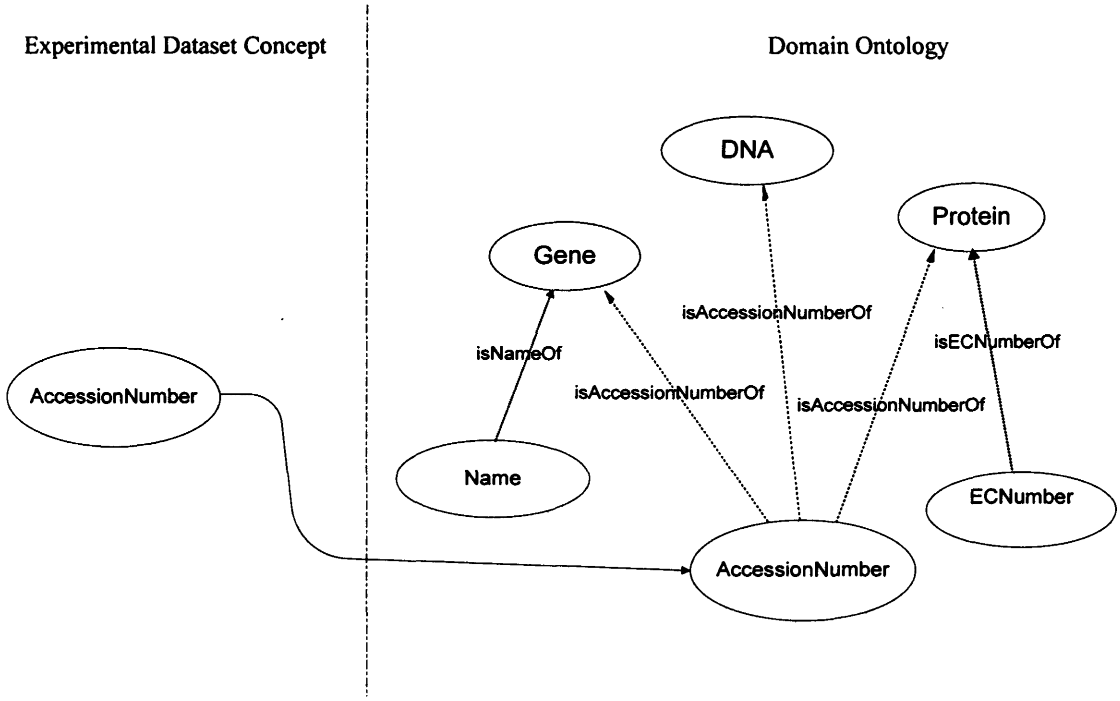


Figure 6.3: Mapping the experimental dataset concept into the Domain Ontology

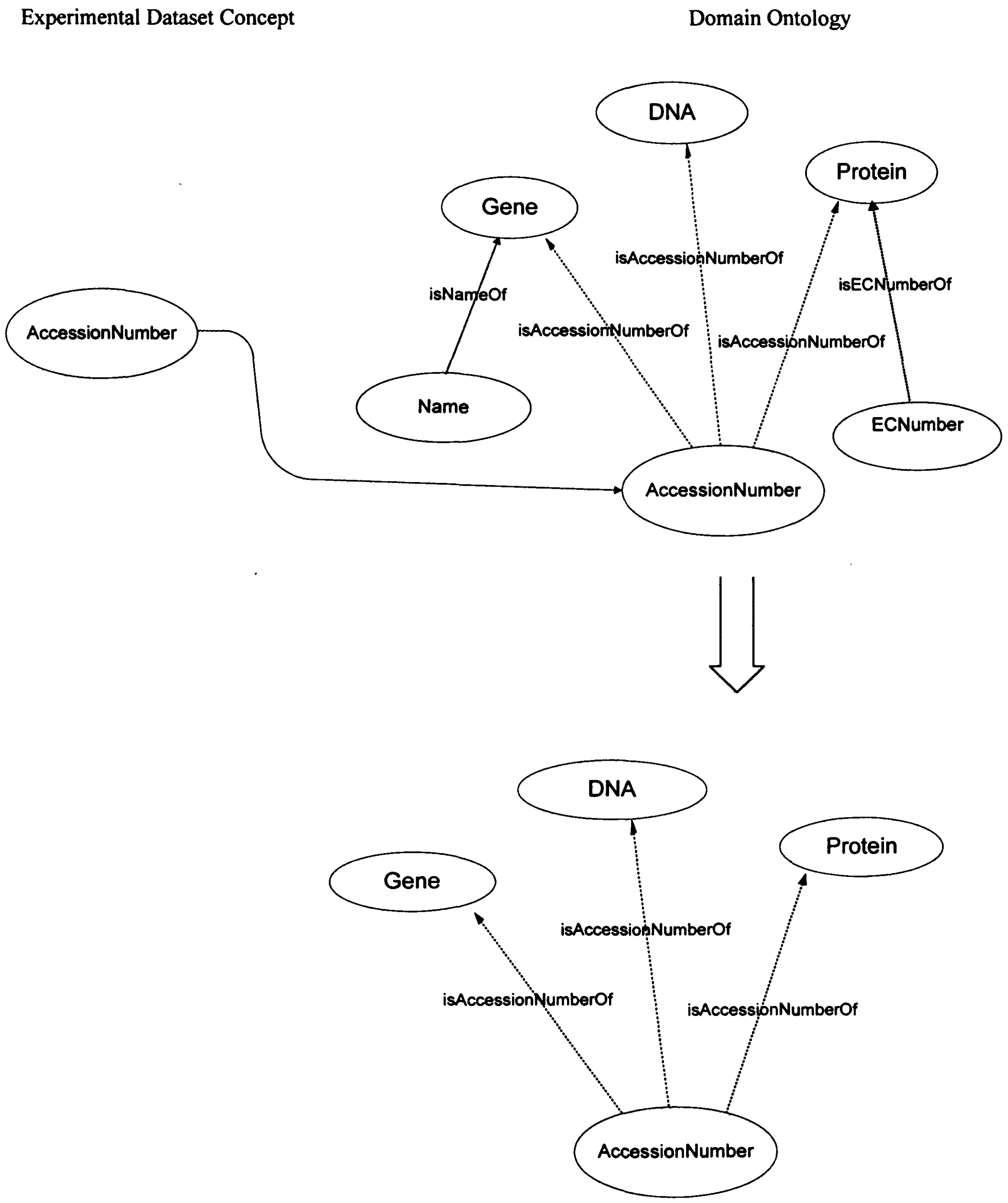


Figure 6.4: Discovered semantic relationships between the experimental dataset concept and domain ontology concepts

6.5 System Architecture

The proposed system consists of the following main components (see Figure 6.5).

Metadata extractor: extracts metadata and has an Application Programming Interface (API) to facilitate interaction between the application and the extractor. It also undertakes the functions of the extractor component, such as experimental file processing and analysis.

Linkage key detector: computes the score for each column of the experimental dataset to identify whether the element is suitable for use as a linkage key to other sources. It uses the column headers and column entries to calculate the scores of each column using formulae 1 and 2 in section 6.3.1.5, and the scoring system in Table 6.1.

Concept mapper: maps the experimental dataset concepts, mainly the candidate key, to the domain ontology's concepts and discovers relationships between them.

Schema creator: creates a schema for the experimental dataset based on the extracted metadata as described in section 6.3.

Data transformer: imports data in an unstructured format and transforms it into a structured format, such as a relational form. It transforms the experimental dataset using the schema created by the schema creator. This creates a populated relational database from an experimental dataset.

Once the experimental dataset is analysed and transformed to a suitable format, it can be linked and integrated with our IDMBD system [10, 12].

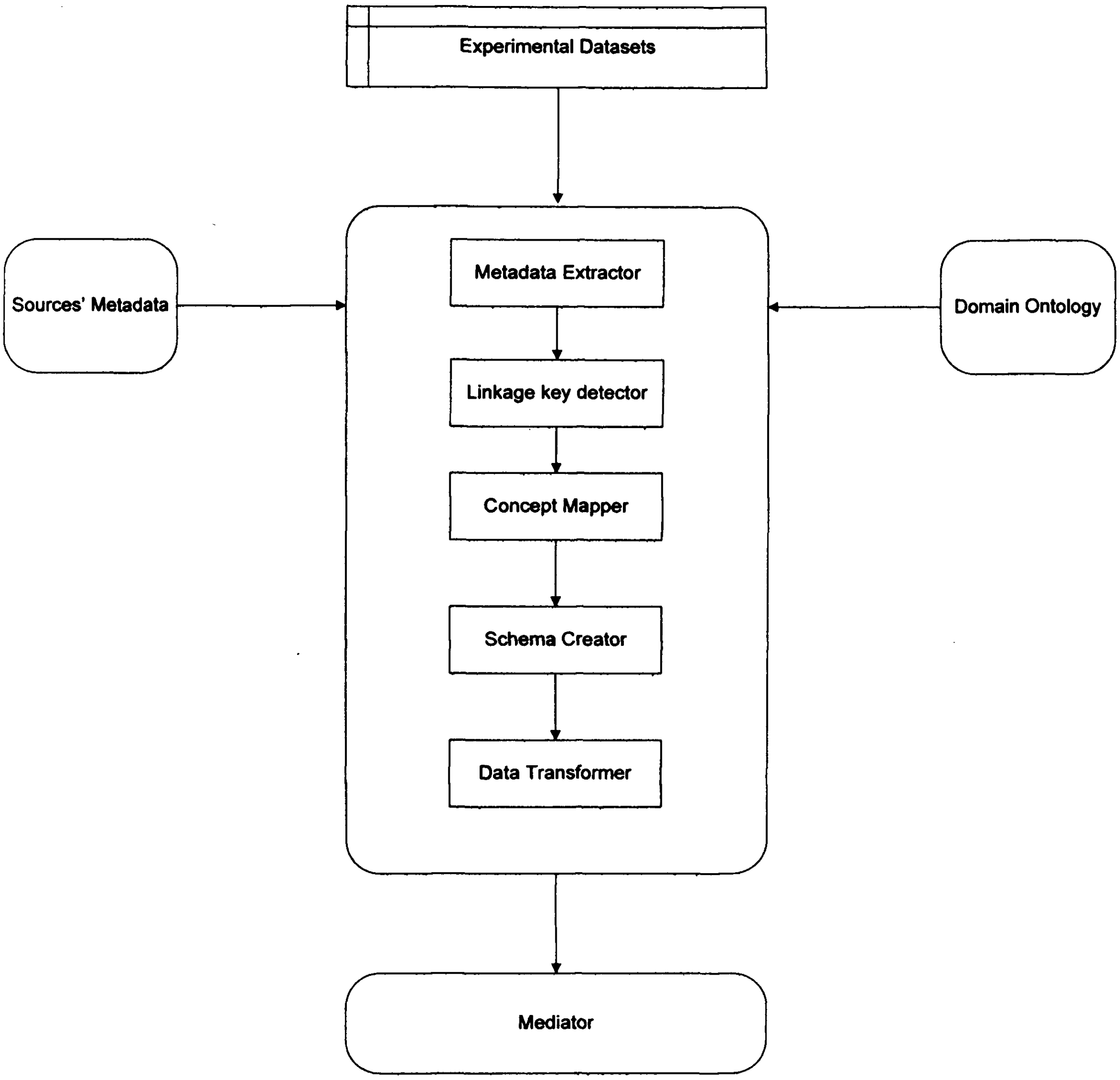
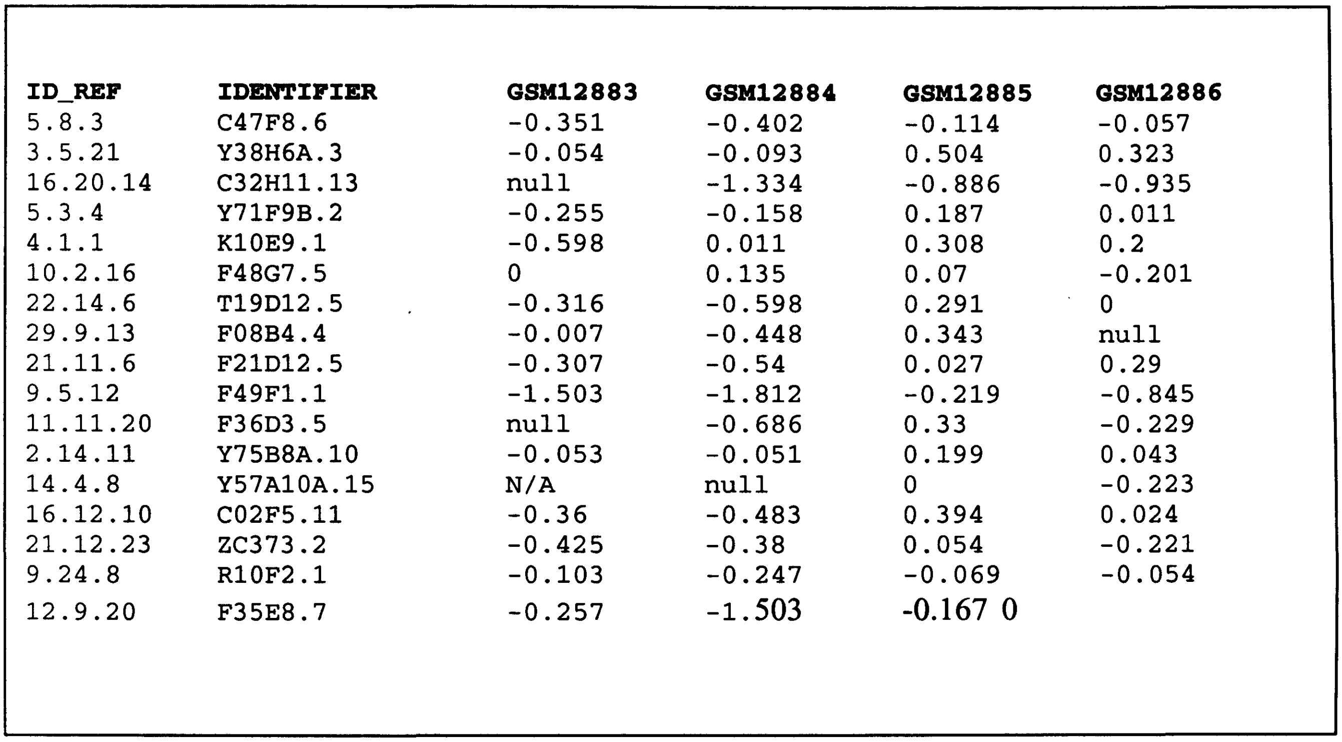


Figure 6.5: Query Handler and Metadata extraction Architecture

6.6 Limitation

Since there is a broad variety of flat file formats, the approach presented in this chapter is not intended to cover all types of flat file formats in their entirety. However, it is a starting point for further enhancements in this direction. The prototype system accepts only delimited flat files, where the first line contains column names or headers (Figure 6.6). However, the principles in the approach can be used also for semi-structured files (e.g. XML) where an element tag can be treated as a heading name. Moreover, some of the principles (candidate key for linkage detection, semantic relationship discovery and ontology mapping) can be used with any file type.



ID_REF	IDENTIFIER	GSM12883	GSM12884	GSM12885	GSM12886
5.8.3	C47F8.6	-0.351	-0.402	-0.114	-0.057
3.5.21	Y38H6A.3	-0.054	-0.093	0.504	0.323
16.20.14	C32H11.13	null	-1.334	-0.886	-0.935
5.3.4	Y71F9B.2	-0.255	-0.158	0.187	0.011
4.1.1	K10E9.1	-0.598	0.011	0.308	0.2
10.2.16	F48G7.5	0	0.135	0.07	-0.201
22.14.6	T19D12.5	-0.316	-0.598	0.291	0
29.9.13	F08B4.4	-0.007	-0.448	0.343	null
21.11.6	F21D12.5	-0.307	-0.54	0.027	0.29
9.5.12	F49F1.1	-1.503	-1.812	-0.219	-0.845
11.11.20	F36D3.5	null	-0.686	0.33	-0.229
2.14.11	Y75B8A.10	-0.053	-0.051	0.199	0.043
14.4.8	Y57A10A.15	N/A	null	0	-0.223
16.12.10	C02F5.11	-0.36	-0.483	0.394	0.024
21.12.23	ZC373.2	-0.425	-0.38	0.054	-0.221
9.24.8	R10F2.1	-0.103	-0.247	-0.069	-0.054
12.9.20	F35E8.7	-0.257	-1.503	-0.167 0	

Figure 6.6: Sample of tab delimited flat file, where the heading names in the first line

6.7 Summary

The process of automatically extracting the metadata from an experimental dataset is an important stage in effectively integrating this dataset with data available in public domain bioinformatics sources. Metadata extracted from this data file can be stored in databases and used to verify data extracted from an experimental dataset. This allows the biologist to keep track of the dataset, and facilitates its future retrieval. The extracted metadata can also be mined to discover useful knowledge. The dataset may also be processed and queried with other bioinformatics data sources to obtain more information. The experimental dataset may contain a number of types of metadata that can be used to add semantic value to the linkage.

This chapter has described an approach for extracting metadata from an experimental dataset. The approach attempts to extract the following types of metadata: element name, type, length and constraints, such as null value allowed and positive value or negative value allowed. The approach was able to identify a suitable linking element to public domain bioinformatics sources.

The approach extracts an experimental metadata and identifies the most suitable linkage key, by a technique based on a mathematical foundation using a proposed scoring system. A domain ontology is also used to mine and discover semantic relationships between an experimental dataset concept and its domain concepts. These relationships are used to enhance the metadata, which helps in linking and integrating the experimental dataset with public domain bioinformatics data sources.

Chapter 7

Implementation

7.1 Synopsis

This chapter describes the implementation of the system presented in Chapter 5 as an illustration of concept of the SLM model. Implementation details of the IDMBD prototype and how relationships between biological objects are used to integrate heterogeneous bioinformatics data sources across species are presented and explained in this chapter. There is an implementation overview, followed by a discussion of the technologies used, and a description of modules. This chapter does not intend to give full details of implementation or a user guide of the system, but rather highlight some of the system's functionality and implementation.

7.2 Requirement Analysis

There are many factors involved in determining the system design and implementation of any system, and the following were important for IDMBD:

- IDMBD should be implemented as an illustration of concept prototype to demonstrate the technique of linkages described in Chapter 4. Other features that exist in other systems, like reports, visualization and integration of bioinformatics analysis tools are beyond the scope of this project.

- The system architecture should be extendible, i.e., it should be

  - capable of allowing new data sources to be added.
  - capable of allowing the addition of new relationships.

- The system should be user driven with respect to the type of relationship and algorithms available to establish relationships with flexibility in setting thresholds.

- The system should be flexible, i.e., it should accept diverse types of experimental data files and different linkages

- The system should be designed in a modular and generic way so that its components can be adapted and reused.

- The architecture should preserve data source autonomy and access up-to-date data.

7.3 Implementation overview

The system framework of IDMBD is composed of a web client layer, web application layer, database connection layer and data sources layer. In this implementation, we chose Apache integrated with Tomcat as the WWW server software, mySql as the database server, and Java language and Java Server Pages (JSP) technique as the means of development except for the Worm wrapper, which is implemented in Perl.

Figure 7.1 illustrates the implementation architecture of IDMBD. The Client GUI interface facilitates user interaction with the other system components in the architecture. Users access the system through a user interface, i.e., Web browser, which accepts a user query, uploads the experimental datasets and displays the results. First, the user sends a request to the web server. Subsequently, the web server transfers the request to the IDMBD, which handles the user request through its modules, which are described in section 7.6.

The IDMBD's mediator interacts with the data sources' wrappers to facilitate the submission of queries and receipt of results. The wrappers thus hide the complexity of the data sources from users and other components. It also interacts with the SLM and Relationships Knowledge Base (RKB) through the Soft Link Adapter (SLA), which makes the SLM appear to the external world to be an object with a set of predefined methods. The web server then returns a response to the client's browsers through the web. The main components in the implementation architecture are:

- **Apache Web Server:** This server is responsible for the services on static HTML pages related to the project and passes JSP requests to the Servlet container, i.e., Tomcat.

- **Tomcat Servlet Container:** This server accepts the incoming Servlet as JSP requests and processes, handles, and responds to them. The JSP pages are simply an interface between the user and the background system. JSP pages let the user enter his/her query, upload experimental datasets and set his/her preference parameters. It then passes this information onto IDMBD.

- **IDMBD modules:** These consist of the main system modules (Section 7.5), Java helper classes and wrapper classes. Java helper classes are responsible for the HTML/XML parsing, processing, data caching and data processing tasks. Wrappers are responsible for the creation, maintenance, and closure of actual database connection classes, the passing of queries to the identified databases, and the receipt of incoming data from the databases.

- **Wrapper Layer:** This layer is designed to be extensible so that, in future, new data sources and connection handlers can be easily and seamlessly inserted into the system. At present, there are two different interfaces because of the two data sources.

  - **JDBC interface:** Our Java classes use this interface like a standard Java Database connection API. We create, maintain, and close the database connection according to

JDBC APIs, and follow the same pattern for query construction and result set processing.

- **AcePerl interface:** This is used to connect to the AceDB. We use its library and follow its interface to create, maintain, and close database connections. Query construction, passing and result set processing are handled according to the AcePerl interface.

- **Data Source Layer:** data sources are invisible to end users and composed of heterogeneous data sources. Currently, the following species- specific data sources are used in our implementation:

  - **MGD:** includes information concerning the genetics, genomics and biology of the mouse.

  - **WormBase:** includes information concerning the genetics, genomics and biology of C. elegans and some related nematodes.

- **RKB**

A Soft Link Model was created between mouse and C. elegans as described in section 4.3 for the following relationships: Homolog, Orthology, Molecular Function, Biological Process, and Cellular Component. First, we parsed all mouse sequences and worm sequences from Swiss-Prot using the parser. Then we used the BLAST algorithm to compute the homology between the sequences. For generating the Molecular Function, Biological Process and Cellular Component, we used the algorithm described in section 4.3.3.4.2. The relationship instances were then stored in a relational table as (source-object identifier, target-object identifier, relationship closeness) in mySql databases. After we had built all relationship tables, we created our SLM and stored it in an XML file as shown in Figure 7.2, which describes the relationships between the concepts of the data sources. The RKB can be used to build protein-protein interactions as lines (edges) forming a network between points (nodes). Data can be visualized as a network graph directly from RKB using visualization software (Figure 7.3).

7.4 Choice of programming language

In implementing the IDMBD prototype, Java was used for most components and Perl was used to create wrappers for AceDB data sources. Java supported object-oriented design, modularity in the system design, easy integration with other Java, C and C++ components and availability of APIs. However, the system can be implemented using any programming language that provides support for developing distributed applications, such as C, C++ or Java. Java was chosen to implement the system, due to the following advantages over other programming languages:

- It is a platform independent language that allows developers to write software that can be compiled once for execution on different platforms.
- Due to Java's current popularity, many developers are familiar with the language and will therefore be able to use our system.
- Several libraries and classes are implemented in Java.

The technologies used are summarised in Appendix C with reasons for use.

7.5 Modules

Figure 7.4 shows the modules of the IDMBD system. These modules were designed to be a generic so they could be adapted and reused. Samples of Java classes are presented in Appendix D. This section provides descriptions of these modules.

7.5.1 Soft Link Model

The Soft Link Model is responsible for discovering relationships between different objects. Six types of relationships are implemented. It also communicates with the mediator during the integration process to enrich query results with additional information from other species based on a relationship of interest. It receives requests from the mediator and then collects data from the SLM metadata and RKB to find possible related data sources and respond to the mediator. Thus, it helps in providing a flexible linkage between data sources using the relationships created.

- **SoftLinkAdapter (SLA):** This module comprises a set of application programming interfaces (APIs) to interact with the mediator. When it receives a request from the mediator to find data related to entries sent by the mediator, it determines whether the relationships exist in the Soft Link Model. If they exist, it fetches them from RKB and returns the related data sources, concepts and identifiers to the mediator. This module has several primitives for IDMBD. A brief description of these primitives follows:

    - getRelatedConcept: the main primitive in SLA, which calls other methods to fetch all related entries from other data sources.
    - getRelation: this gets the relationships for a specific concept in a specific data source from SLM metadata. It retrieves the relationship name, concept and the data source name for related sources.
    - getRelations: this gets all relationships existing in a specified Soft Link Model.
    - GetMatchEntriesInDataSource: returns related entries from other data sources. It calls the wrapper manager to fetch records from related data sources, and retrieves a record from a data source that has relationships with specified entries.

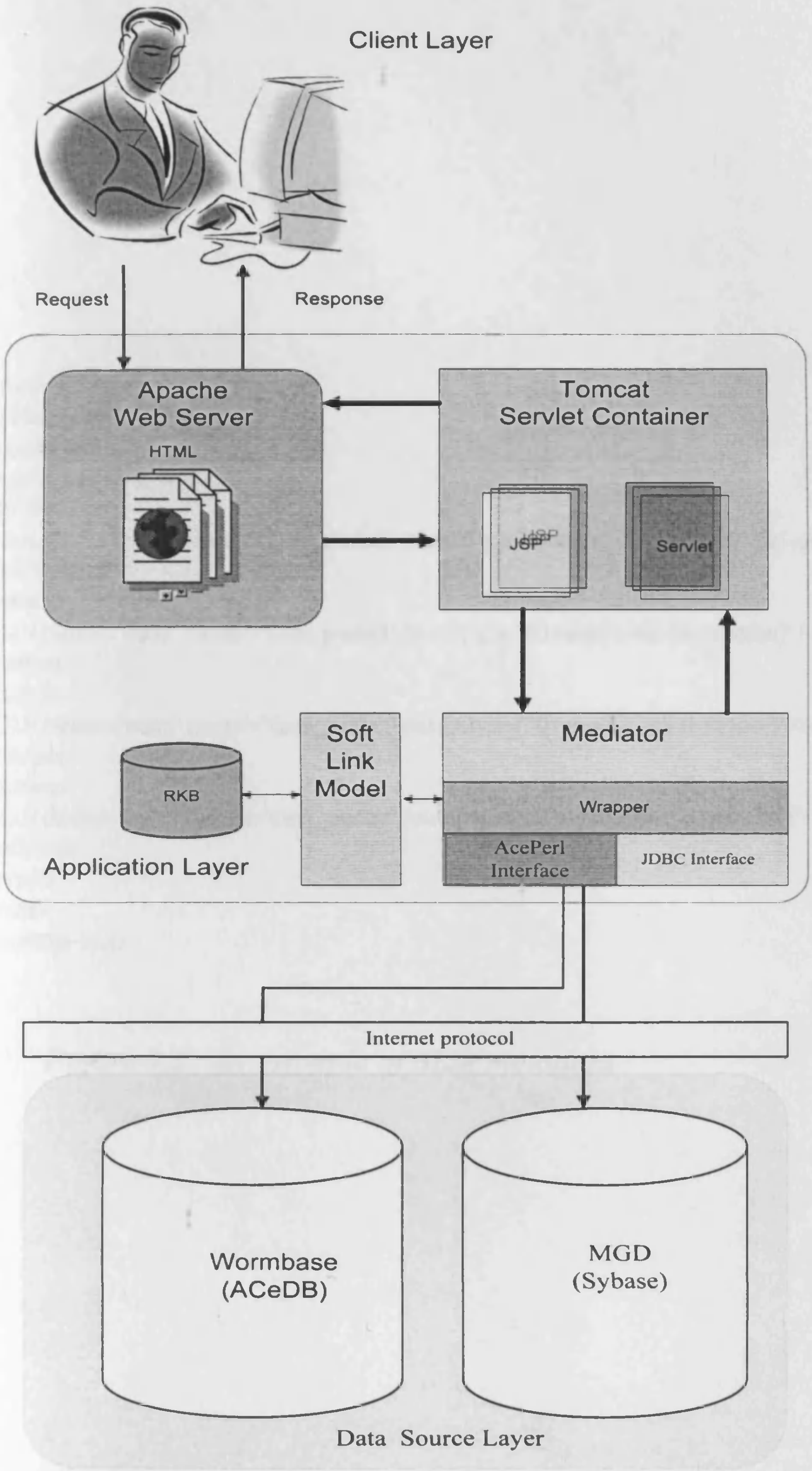
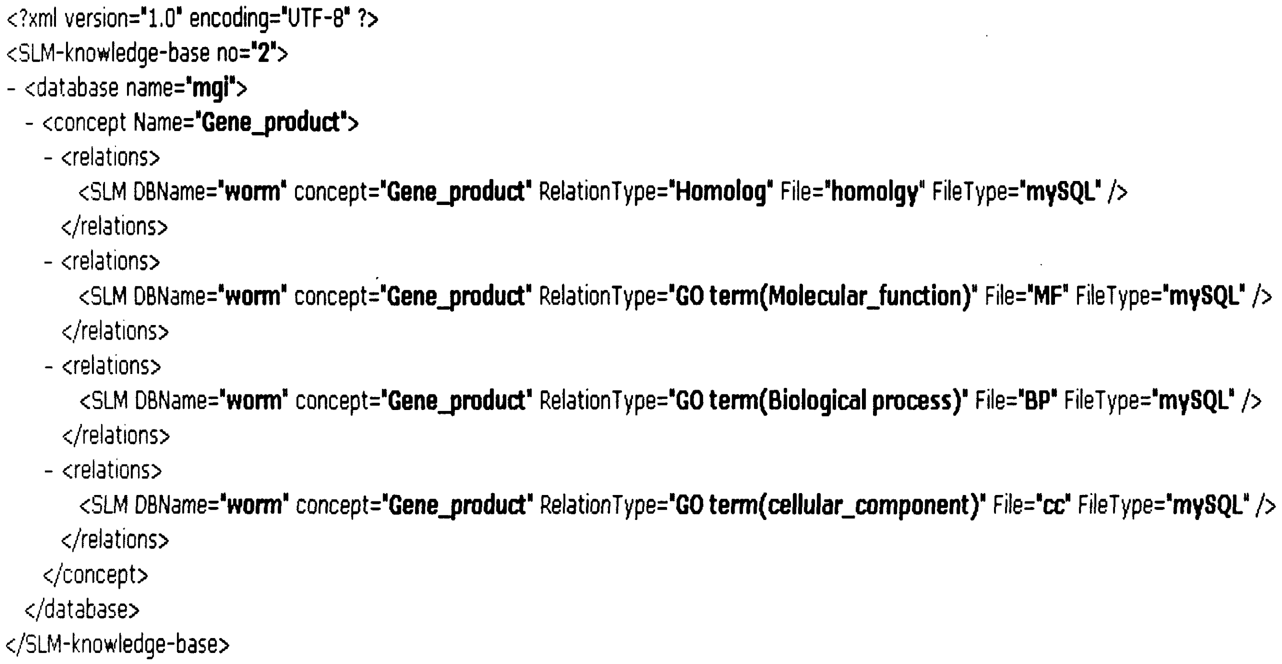


Figure 7.1: An overview of the implementation architecture



```
<?xml version="1.0" encoding="UTF-8" ?>
<SLM-knowledge-base no="2">
- <database name="mgi">
  - <concept Name="Gene_product">
    - <relations>
        <SLM DBName="worm" concept="Gene_product" RelationType="Homolog" File="homolgy" FileType="mySQL" />
      </relations>
    - <relations>
        <SLM DBName="worm" concept="Gene_product" RelationType="GO term(Molecular_function)" File="MF" FileType="mySQL" />
      </relations>
    - <relations>
        <SLM DBName="worm" concept="Gene_product" RelationType="GO term(Biological process)" File="BP" FileType="mySQL" />
      </relations>
    - <relations>
        <SLM DBName="worm" concept="Gene_product" RelationType="GO term(cellular_component)" File="cc" FileType="mySQL" />
      </relations>
    </concept>
  </database>
</SLM-knowledge-base>
```

Figure 7.2: An example of SLM metadata

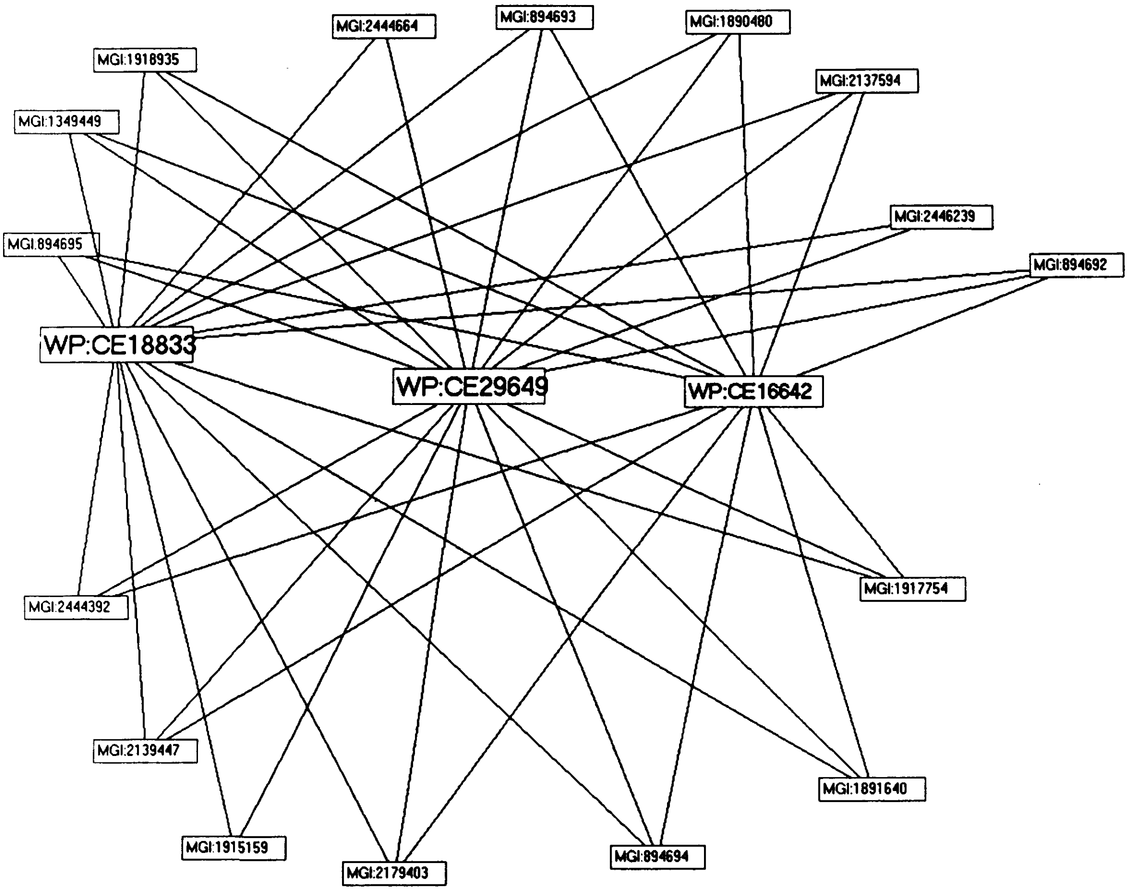


Figure 7.3: A graph represents protein-protein relationships between mouse and C.elegans. Each rectangle represents a different protein and each line indicates that the two proteins have relationships. Only a very small set of RKB is visualized here

  - GetMatchEntriesInRelationTable: fetches relationship table entries that match the specified identifier, using the Relationship wrapper interface, it retrieves all records related to the specified identifier and data source.

- **GenerateSoftLinkTable:** is used to create a soft link model to discover semantic relationships between concepts across data sources. This generates homology, orthology, paralog, Molecular Function, Biological Process and Cellular Component relationships between genes. Different algorithms are used to calculate relationship closeness between objects. The SLM uses a mySQL database to store relationship instances whenever there is a relationship between a pair of entries in a pair of data sources.

- **buildSLM**: is responsible for creating SLM metadata and storing relationships in RKB.

7.5.2 Configuration

This module is responsible for registering new data sources to the system as well as loading configuration files on the system execution. It consists of two main sub modules:

- **Register:** registers new data sources within the IDMDB system, by specifying data source information: name, location, wrapper, and schema, type of data source and access procedures that can be used to interact with a data source (Appendix B).

- **Config:** parses the configuration file and loads all registered data source wrappers and Soft Link Models on the system execution. The configuration file "conf.sys" contains registered data sources, their wrapper classes, and available soft link models.

7.5.3 Mediator

Mediator plays intermediate roles between users and data sources. It also communicates with the soft link module to retrieve relationships between data sources. This module consists of sub modules:

- **Wrapper Manager (WM):** instantiates the various data sources' wrappers the system is configured to use. It manages existing wrappers and performs other tasks: loading existing wrappers, communicating with the Soft Link Model to retrieve relationships between objects across sources, and removing duplicates during result assembly from different sources. It invokes an appropriate wrapper to get responses from sources. It loads both the Perl and Java wrapper module on demand dynamically. It uses data sources' wrappers to access other objects from those data sources. The WM module consists of an Application Programming Interface (API) to facilitate interaction between the application and the data sources' wrappers as well as wrapper loading.

- **Query Handler:** plays a major role in integrating the expression dataset with the public bioinformatics data sources. It is responsible for linking the metadata with the domain ontology, detecting the suitable linkage key and extracting metadata from the gene expression data set. Query Handler has several API primitives; a brief description is offered in Table 7.1.

- **Source Selection:** is responsible for selecting the appropriate data source to answer a user query. User requests received by the Web server module are forwarded to the source selection module to decide which data source is appropriate. The data source is selected based on the user query and on how many relationships are associated with the data source.

7.5.4 Wrapper

The wrapper module is a class with specific entry points that provides access to a class of data sources. The wrapper uses the standard connection API of the data sources. For example, it uses the JDBC driver to connect to a relational database, and retrieves data, and uses AcePerl to connect to the ACEDB data source. Specific wrappers are necessary for each data source integrated into the mediator. Several wrappers are implemented into IDMBD. These wrappers are:

- **MGI_Wrapper:** is a Java class that implements the Wrapper interface and is loaded into the WM. It uses the JDBC driver to connect to the MGI data source, which is a relational database, and to retrieve data.

- **Worm_Wrapper:** is a Java class that also implements the Wrapper interface and is loaded into the WM. It uses the AcePerl driver to connect to the AceDB and to retrieve data.

- **Relationship_Wrapper:** The Relationship Wrapper class implements the wrapper interface, which provides methods to load JDBC drivers, establish new database connections, and fetch relationship instances between data sources. This wrapper is invoked by the SoftLinkAdapter (SLA) to fetch instances from RKB.

- **GO_Wrapper:** provides an interface to the Gene Ontology (GO). It uses JDBC drivers to access GO database.

- **UniGene_Wrapper:** provides an interface to Unigene. It uses JDBC drivers to access the UniGene database.

7.5.5 Parser

This module is responsible for parsing BLAST output, XML files and DNA and amino acid sequences. There are three parsers implemented in the system:

- **SLMParser**: parses the SLM file and gets all the relationships from the Soft Link Model and loads them into a hash table for later use.

- **BlastParser**: is used to parse the BLAST output and extract the sequence similarity score between each pair of sequences and the identity percentage. This module uses the BioJava Blast-like parsing framework, which allows direct SAX2-like parsing of the native output from Blast-like bioinformatics software (bioJava.org). It uses *BlastLikeSAXParser* and *SeqSimilarityAdapter* of the biojava project [130].

- **SequenceParser**: is used to parse sequences.

7.5.6 UserInterfaces

We developed two interfaces:

- **End-user interface**: a web-based interface for bioinformatics data source integration based on the built prototype SLM. It facilitates access to other system components in the system. It is used for data integration and to link experimental datasets with data available in public data sources.

- **Maintenance interface:** for the administrator who uses it to reveal relationships between concepts within data sources, i.e., it is the main user interface that allows the administrator to register new data sources in the system and build a Soft Link Model between concepts in data sources and create RKB.

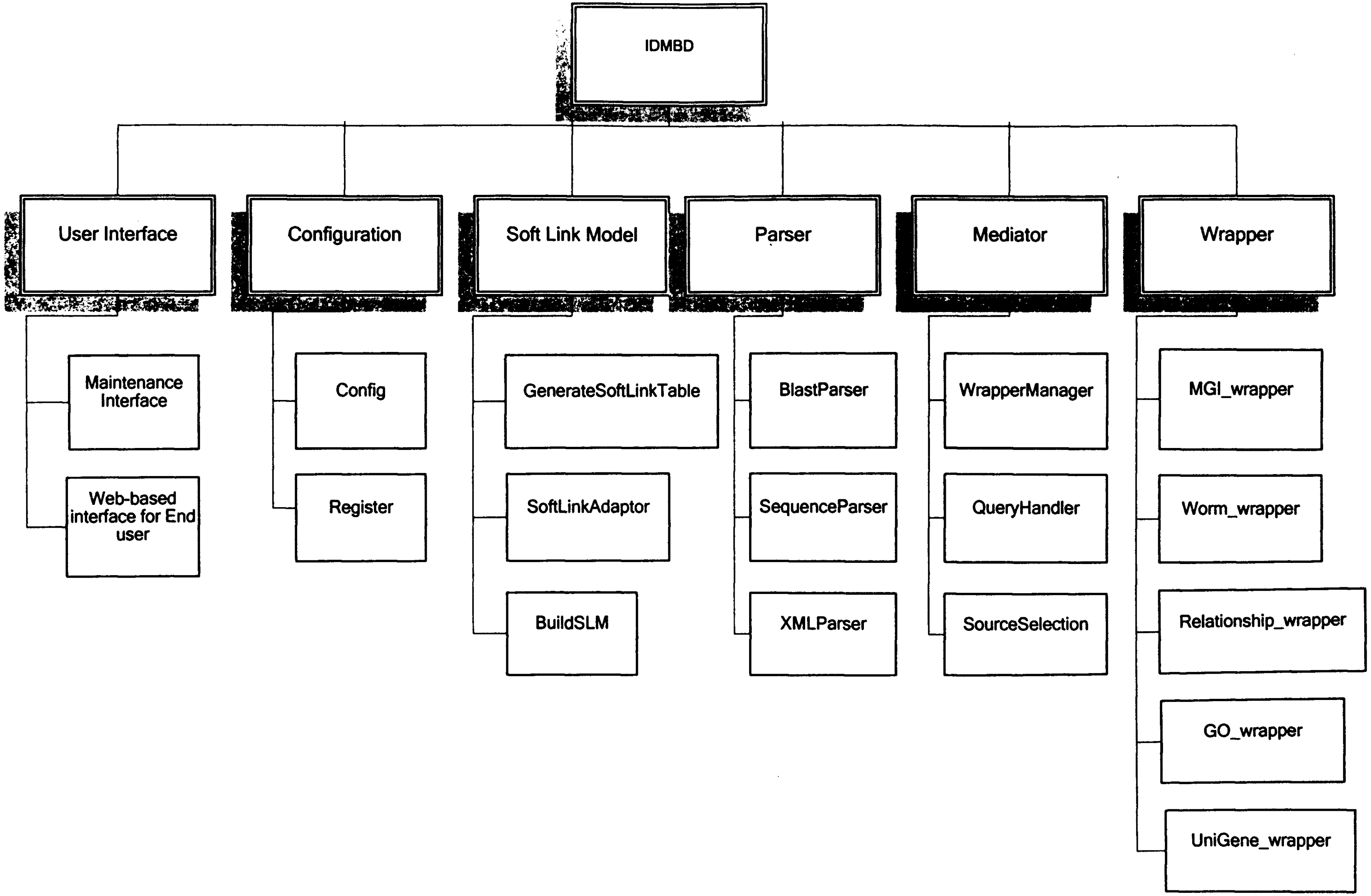


Figure 7.4: The IDMBD modules

Method	function
ExtractMetadat	returns a list of elements' names of an experimental dataset file.
isUnique	returns true if the element value is unique across the dataset.
isSingleValue	returns true if the element value across the dataset has a single value.
isKey	returns true if the element's name includes one of the following words: key, accessio identifier, number, id, and no.
isAmbiguous	returns true if the element has at least an ambiguous value.
isNull	returns true if an element has at least one null value otherwise true.
DataType	returns the data structure type of an element.
elementLength	returns the maximum length of an element on the dataset.
ComputeScore	returns the score for each element.
isDate	checks the selected element to make sure the value contained appears to be a vali date. If the value does not appear to be a valid date, then the column type will not consider a date.
isString	tests the selected element to make sure that it contains a string (contains onl characters A-Z and a-z.)
isDouble	tests the selected element to make sure that it contains a double value (contains only numbers 0-9 and a decimal point.)
isIntger	tests the selected element to make sure that it contains a numeric value. It does this b passing the string into the parseInt() function.

Table 7.1: Query Handler methods

7.5.6.1 Relationship discovery

When an administrator (who does the relationship discovery) executes the system, he/she is presented with the GUI main window shown in Figure *7.5*. This has the following functions:

- ***Registration***: This function is used to call the register module to add a new data source.

- ***Parser:*** This function is used to call the parser module.

- ***Relationship Discovery***: This function is used to invoke the *GenerateSoftLinkTable* module to create new relationship instances and save relationship instances on a RKB.

- ***Soft Link Model***: This function is used to call *BuildSLM* to create a New Soft Link Model, builds the SLM metadata and stores it in a XML file. It is also used to add a new entry to the Soft Link Model.

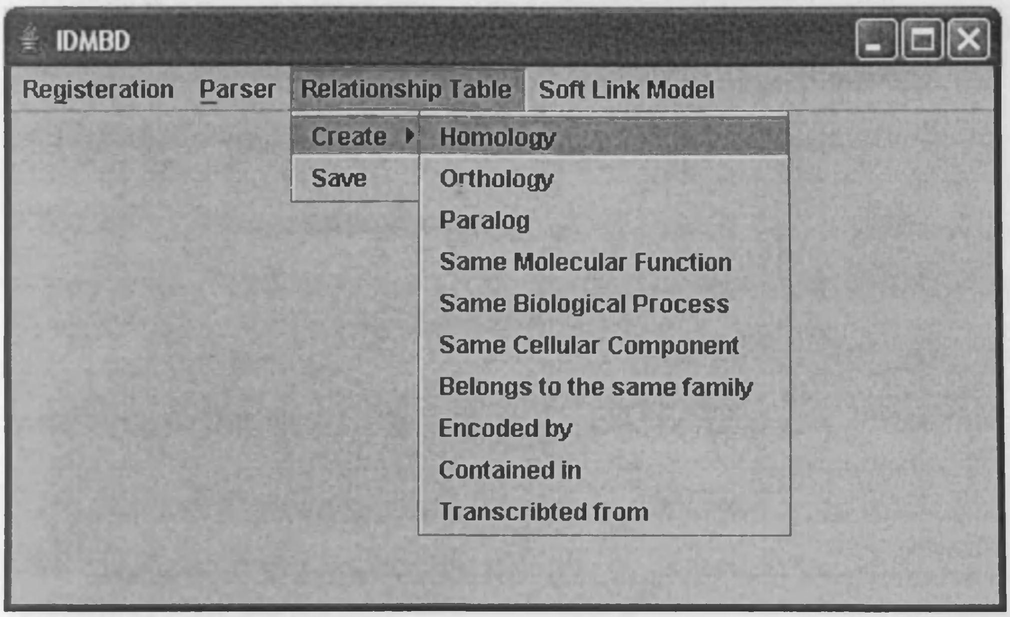


Figure 7.5: GUI Main interface for relationship discovery and building SLM

Building of the RKB is performed through an automatic process. RKB is generated by using algorithms to calculate relationship closeness of interest between objects across data sources. Metadata about the relationships is stored in SLM metadata in XML format. The RKB is built in a bottom-up fashion by adding and merging incrementally the instances of objects that have relationships. This is done by choosing the *Relationship Table* option from the main menu and choosing the relationship type to be created between concepts of data sources. The user will be prompted with an interface as in Figure 7.6. This interface allows the user to specify:

(1) a pair of data sources to be involved in the relationships discovery.

(2) a pair of concepts of data sources to be involved in the relationships discovery.

(3) an algorithm to compute the degree of relationships or similarity between the properties of the concepts.

Once the discovery process is finished, the user can browse discovered relationship results in a separate window, and save them into the RKB and SLM metadata.

7.5.6.2 Integration Process

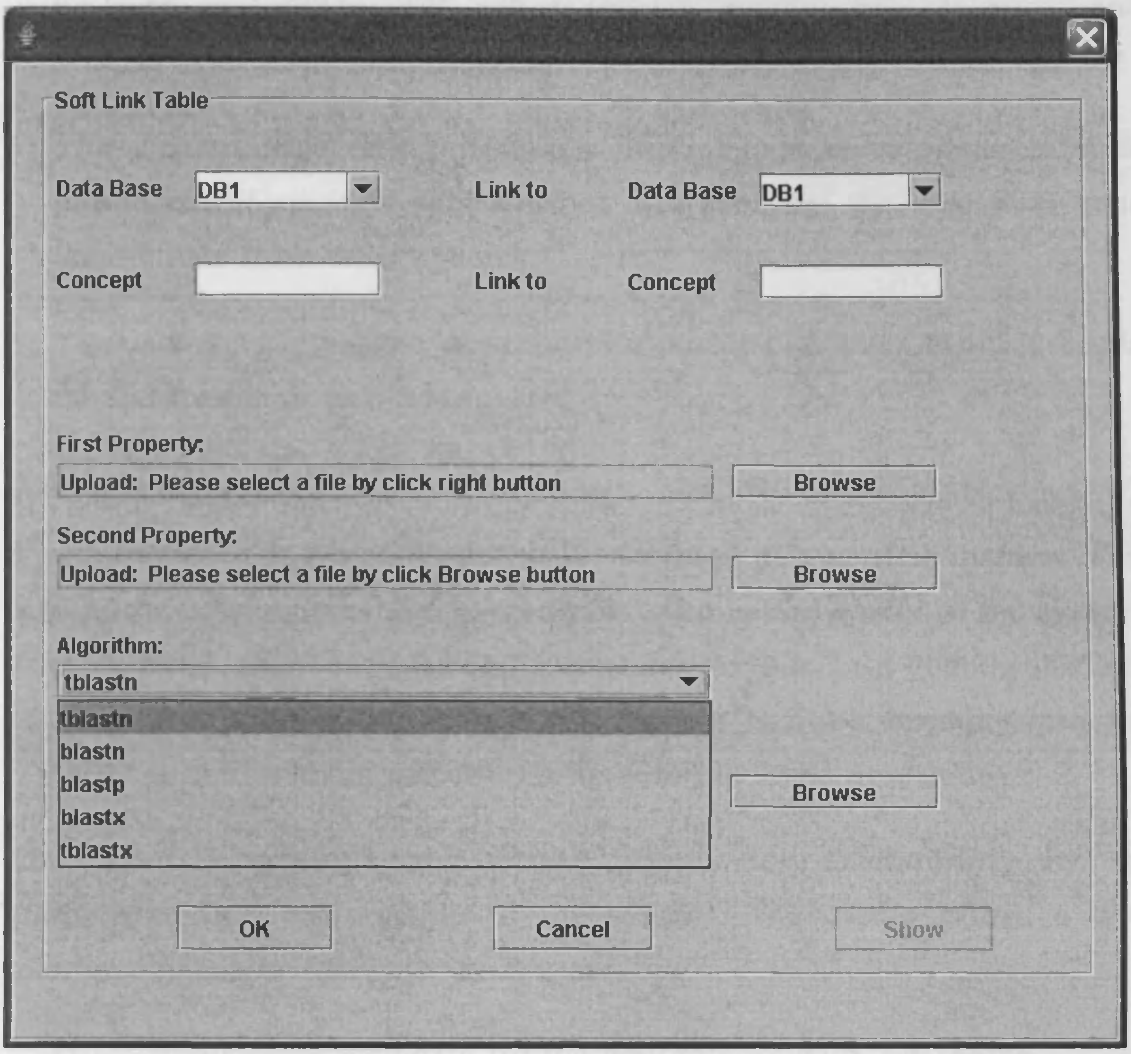
When a user executes the GUI she/he is presented with the GUI main window shown in Figure 7.5. This has five options:

- ***Overview***: gives a description of the system.
- ***Search Database***: used for single queries.
- ***Advance Search***: used for linking several experimental datasets and comparative genomes and for integrating with public data sources.
- ***Soft Link Model***: used for browsing relationships.
- ***Data Sets Comparisons***: used for cross-species comparisons and comparative genomes.

In the following, we describe briefly, the steps to link experimental datasets with bioinformatics data sources.

1) User chooses the *Advance Search* option from the main menu.

2) User uploads experimental datasets from a file in a flat format through the user interface as shown in Figure 7.8.



Soft Link Table

Data Base DB1 Link to Data Base DB1

Concept Link to Concept

First Property:

Upload: Please select a file by click right button

Second Property:

Upload: Please select a file by click Browse button

Algorithm:

tblastn

Figure 7.6: User interface for discovering relationships between concepts. The user chooses the concepts, data sources and relationships type and the algorithm to compute relationships' closeness

3) IDMBD is used to parse the file and extract metadata of the file as described in Chapter 6. The extracted metadata is shown to the user as illustrated in Figure 7.9 . A candidate key for linkage is highlighted. It is up to the user to decide whether the key recommended by the system or another key from the displayed metadata will be used.

4) The user is prompted by the interface (see Figure 7.10) to set his/her parameters: namely required fields to be retrieved, relationship type to

be used in linkage with other species, relationship closeness and species of experimental datasets.

5) The system links and integrates these experimental datasets with public bioinformatics data sources and provides the user with gene annotations from other species.

6) The user can browse the existing relationships between data sources in tabular format or as a tree format.

7.6 Genericity

A requirement for system is that it is designed in a generic fashion. This is to allow new sources and algorithms to be added easily to the system. The Mediator, SLM module and Parser are written in a generic fashion, so that new sources and relationship types can be brought into the IDMBD system without affecting or needing to write new code.

The IDMBD system's architecture allows for extendibility by the addition of new relationship to the system. With little effort, a new relationship can be added to the system, by:

i) Registering metadata for the new relationship, i.e., name.

ii) Writing or obtaining the necessary algorithm from an internal or external source.

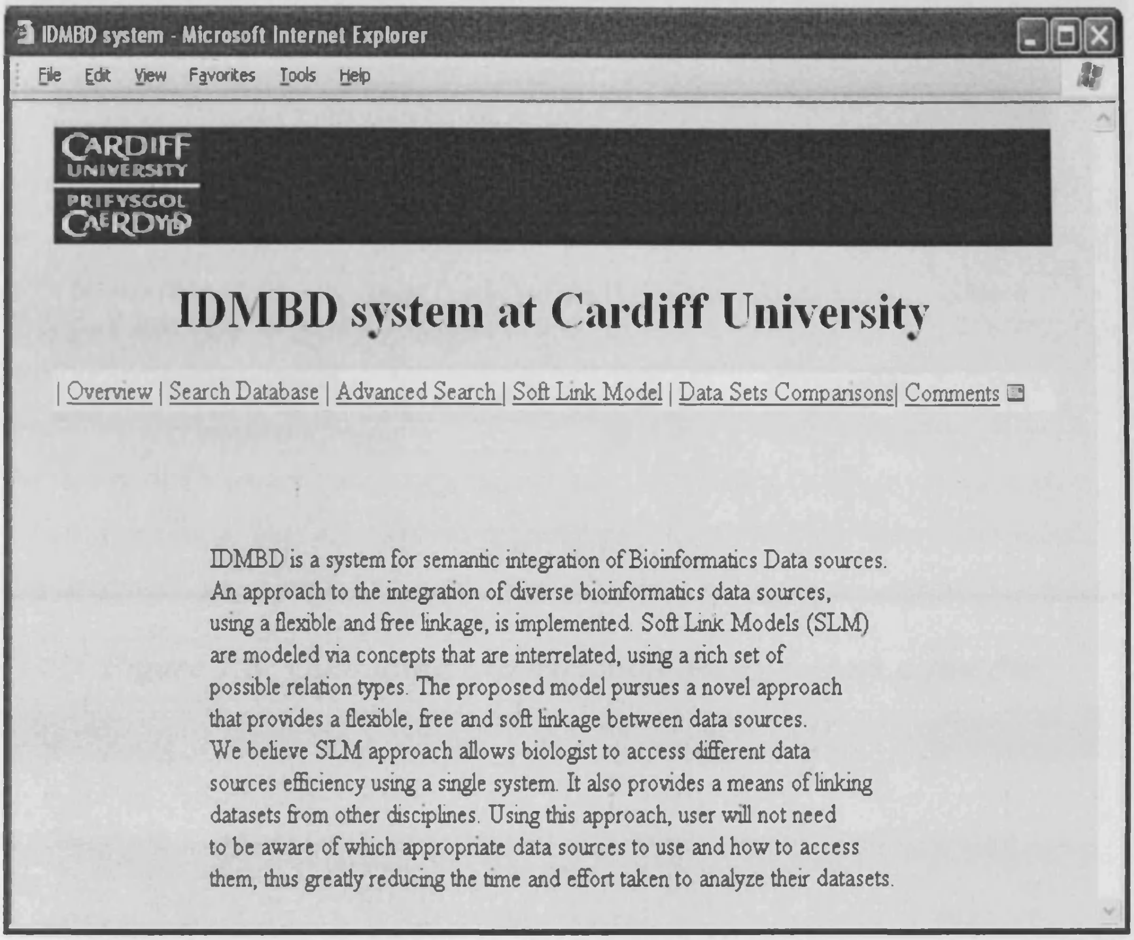
iii) Storing the algorithm's metadata, i.e., name, location, syntax.

iv) Invoking algorithms to mine the data sources for the relationship and measure the relationship closeness between objects in data sources.

v) SLM is built and the relationship tables are generated and added to the RKB.

7.7 Summary

In this chapter, the design and implementation issues of IDMBD were looked at. This chapter began by presenting the requirements. Then, the implementation overview and IDMBD architecture were presented. The choice of programming language and technologies used were introduced also. The description of the modules and components of the IDMBD were presented including the mediator, Soft Link Model, wrappers, configuration, parser and user interface. Finally, snap shots of system menus and interface were presented.

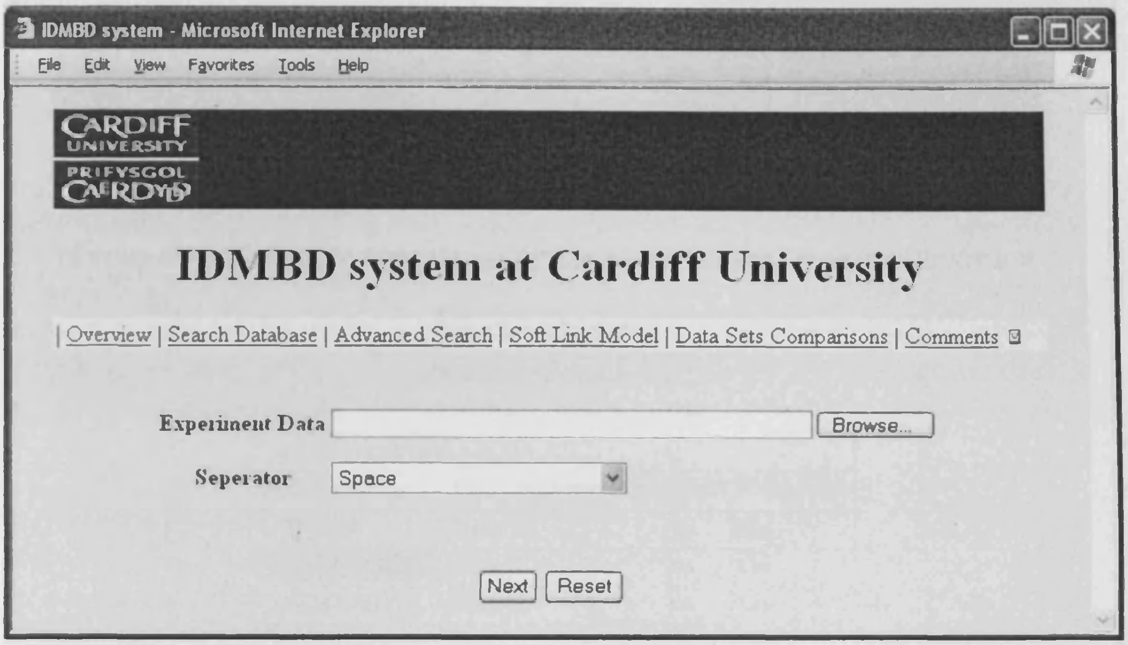

IDMBD system - Microsoft Internet Explorer
File Edit View Favorites Tools Help
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

IDMBD system at Cardiff University

| Overview | Search Database | Advanced Search | Soft Link Model | Data Sets Comparisons| Comments

IDMBD is a system for semantic integration of Bioinformatics Data sources. An approach to the integration of diverse bioinformatics data sources, using a flexible and free linkage, is implemented. Soft Link Models (SLM) are modeled via concepts that are interrelated, using a rich set of possible relation types. The proposed model pursues a novel approach that provides a flexible, free and soft linkage between data sources. We believe SLM approach allows biologist to access different data sources efficiency using a single system. It also provides a means of linking datasets from other disciplines. Using this approach, user will not need to be aware of which appropriate data sources to use and how to access them, thus greatly reducing the time and effort taken to analyze their datasets.

Figure 7.7: snap shot of main web-page interface


IDMBD system - Microsoft Internet Explorer
File Edit View Favorites Tools Help
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

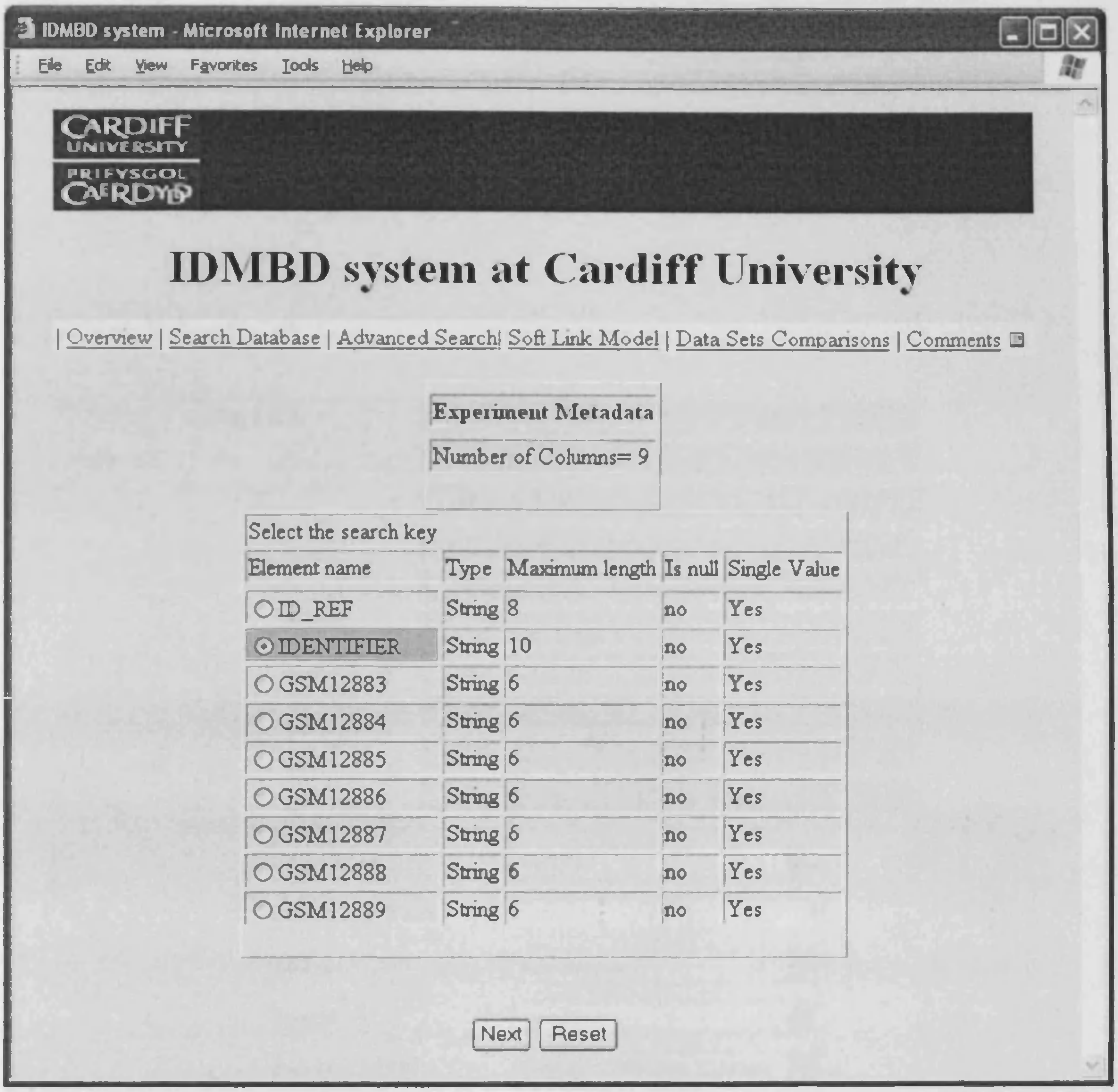
IDMBD system at Cardiff University

| Overview | Search Database | Advanced Search | Soft Link Model | Data Sets Comparisons | Comments

Experiment Data [ ] Browse...

Seperator Space

Figure 7.8: Uploading experimental data set from a flat file


IDMBD system - Microsoft Internet Explorer
File Edit View Favorites Tools Help
CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

IDMBD system at Cardiff University

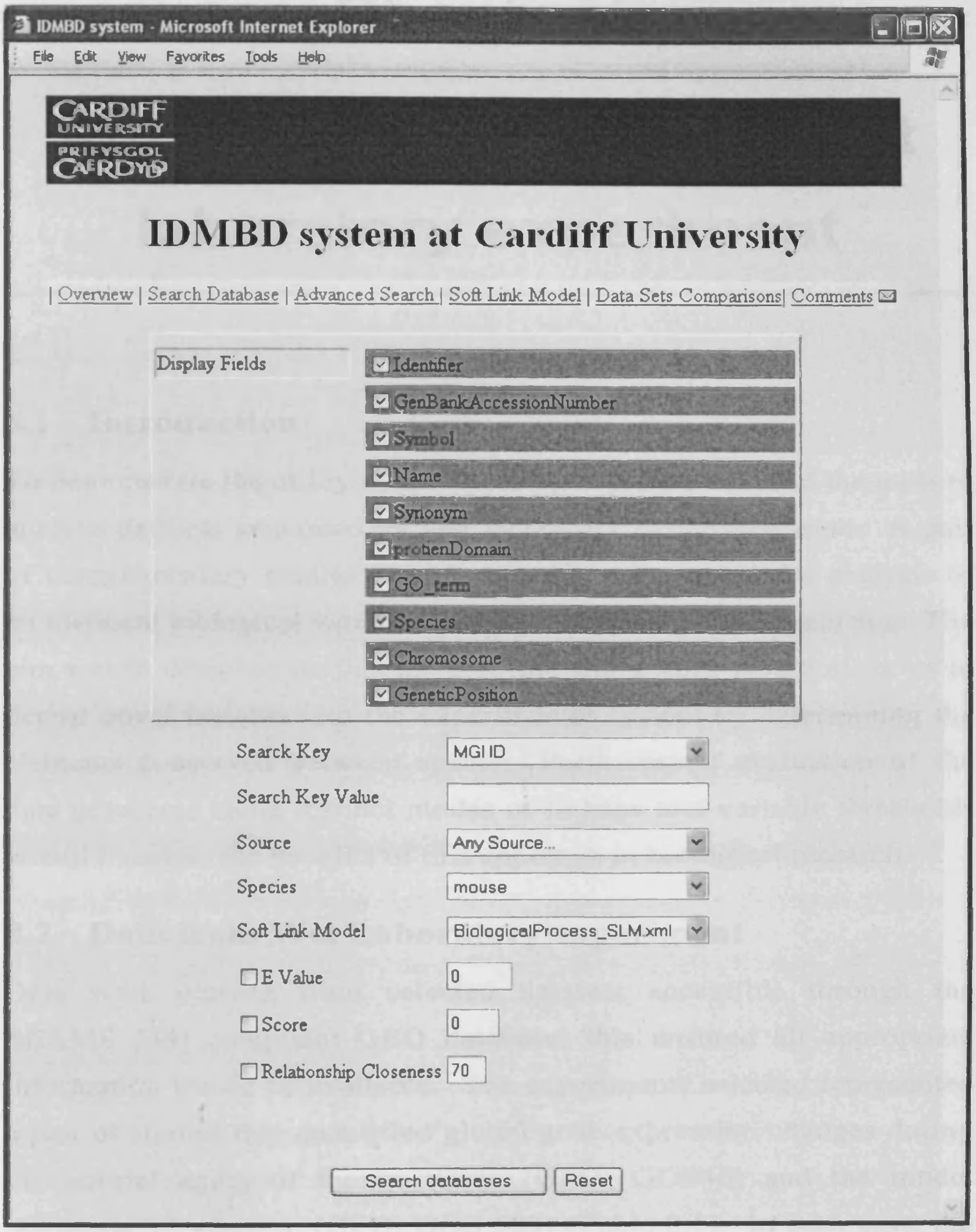
| Overview | Search Database | Advanced Search| Soft Link Model | Data Sets Comparisons | Comments

Experiment Metadata
Number of Columns= 9

Select the search key

Element name	Type	Maximum length	Is null	Single Value
○ ID_REF	String	8	no	Yes
● IDENTIFIER	String	10	no	Yes
○ GSM12883	String	6	no	Yes
○ GSM12884	String	6	no	Yes
○ GSM12885	String	6	no	Yes
○ GSM12886	String	6	no	Yes
○ GSM12887	String	6	no	Yes
○ GSM12888	String	6	no	Yes
○ GSM12889	String	6	no	Yes

Figure 7.9: The metadata detected from experimental data set. The candidate linkage key is highlighted


IDMBD system - Microsoft Internet Explorer
File Edit View Favorites Tools Help
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD
| Overview | Search Database | Advanced Search | Soft Link Model | Data Sets Comparisons| Comments
Search databases
Reset

IDMBD system at Cardiff University

Display Fields

- ☑ Identifier
- ☑ GenBankAccessionNumber
- ☑ Symbol
- ☑ Name
- ☑ Synonym
- ☑ protienDomain
- ☑ GO_term
- ☑ Species
- ☑ Chromosome
- ☑ GeneticPosition

Searck Key	MGI ID
Search Key Value	
Source	Any Source...
Species	mouse
Soft Link Model	BiologicalProcess_SLM.xml
☐ E Value	0
☐ Score	0
☐ Relationship Closeness	70

Figure 7.10: Schema view and user parameters for integration process

Chapter 8

Analysis of data from a wet laboratory experiment

8.1 Introduction

To demonstrate the utility of our prototype system, we used the tools to analyse datasets generated by wet laboratory experimentation. A pair of complementary studies was chosen that represented the analysis of an identical biological variable studied in two different organisms. The aim was to demonstrate that the soft link framework would allow us to derive novel insights into the experimental system by determining the elements conserved between species. Furthermore, evaluation of the data generated using distinct modes of linkage and variable thresholds would illustrate the benefits of this approach in biological research.

8.2 Data from Wet Laboratory experiment

Data were derived from selected datasets accessible through the MIAME [39] compliant GEO database; this ensured all appropriate information would be available. The experiments selected represented a pair of studies that quantified global gene expression changes during the normal aging of mouse tissue (GEO: GDS40) and the model nematode, *C. elegans*. (GEO: GDS 583) (Table 8.1). In each case, the researchers conducting the primary experiments derived a cohort or set of genes that showed statistical age-related changes in their expression pattern. A set of 500 age-related genes were identified in mouse[1], whilst the nematode experiments yielded approximately seven times that number (3534) [148]. This difference may stem from the fact that the mouse study was targeted at a specific tissue, involved in cardiac development, whilst the nematode experiment derived changes from the whole organism.

8.3 Objectives of the SLM Analysis

Through the analysis of these datasets, we aimed to evaluate the significance of altering both the method and threshold of linkage within the SLM used when determining cross-species conservation. This process should have allowed us to determine the optimal threshold for a cross-species orthology relationship. Also by defining the intersection between elements conserved by orthology and ontological classification, this analysis might focus future laboratory studies on key elements of the aging process. The initial step required to achieve these objectives was to integrate experimental datasets with the primary data sources for the two species in question, MGI for mouse and Wormbase for *C. elegans*. We reanalyzed the data several times, altering various parameters (relationship types and threshold) to generate unique groups of gene objects conserved between the experimental datasets under the different relationships. In turn, these lists were analysed for intersections indicating molecular elements closely linked to the biological variable being studied. The sets generated were analyzed to demonstrate whether they provided a functional enrichment over the original base datasets.

	MOUSE	**NEMATODE**
Accession	GDS40	GDS583
Title	Cardiac development, maturation and aging	Aging time course, normal adult
Data set type	gene expression array-based (RNA / in situ oligonucleotide)	gene expression array-based (RNA / spotted DNA/cDNA)
Dataset size	Up regulated genes: 500 genes	Up regulated genes: 3534 genes
Species	mouse [Mus musculus]	nematode[C. elegans]
Summary	Benchmark gene expression profile of heart ventricle at various ages to monitor changes in cardiac development. Examined embryonic stages through adolescence and adulthood.	Examination of normal adult aging using synchronized populations at 0 - 144 hours. Employed CF512 fer-15(b26) II; fem-1 (hc17) IV mutant strain, which has defective spermatids thus eliminating contributions from embryonic transcripts.

Table 8.1: Comparison of the experimental metadata describing the two wet lab experiment used for SLM analysis

8.4 Integration of Wet Laboratory data into "Soft Link Model Environment"

A high-level schematic overview of query workflow is given in Figure 8.2 and illustrates how the various inputs and outputs are interlinked. The phases of analysis include the following stages:

8.4.1 Metadata extraction

The initial stage of the experiment exploits the flat file representation of the experimental datasets. The system extracts the metadata data from this file and recommends a key for linkage; in this example, the recommended linkage key for the mouse data, highlighted in Figure 8.1, is the *GenBank ID.* User defined, additional metadata can be extracted from the original data file as shown in Figure 8.1.

8.4.2 Identifier conversion

The system subsequently maps the dataset linkage keys to specific-species identifiers, MGI ID and WP Protein ID, for the mouse and *C. elegans,* respectively. A network of complex relationships may be utilised to accomplish this mapping. For example, MGD links to GenBank either through the field "Markers" (in the "genes" table) or field "molecular probes" or "segments" (for anonymous DNA segments):

> *Relationship 1 (R1):* GenBank Accession -> Marker (gene).
> *Relationship 2 (R2):* GenBank Accession -> Marker (gene), GenBank Accession ->probe, Probe-> Marker (gene)
> *Relationship 3 (R3):* GenBank Accession --> Marker (gene), GenBank Accession->probe/segment, Probe->Marker(gene)->GenBank Accession ->UniGene identifier, UniGene identifier -> Marker (gene).

8.4.3 Cross species transformations

The system uses specific-species identifiers together with pre-calculated relationship tables (RKB) to transform the gene lists from one species to their counterparts, as defined by the function of the relationship table and the threshold under which it is sampled in another species. This transformation is central to the SLM processes.

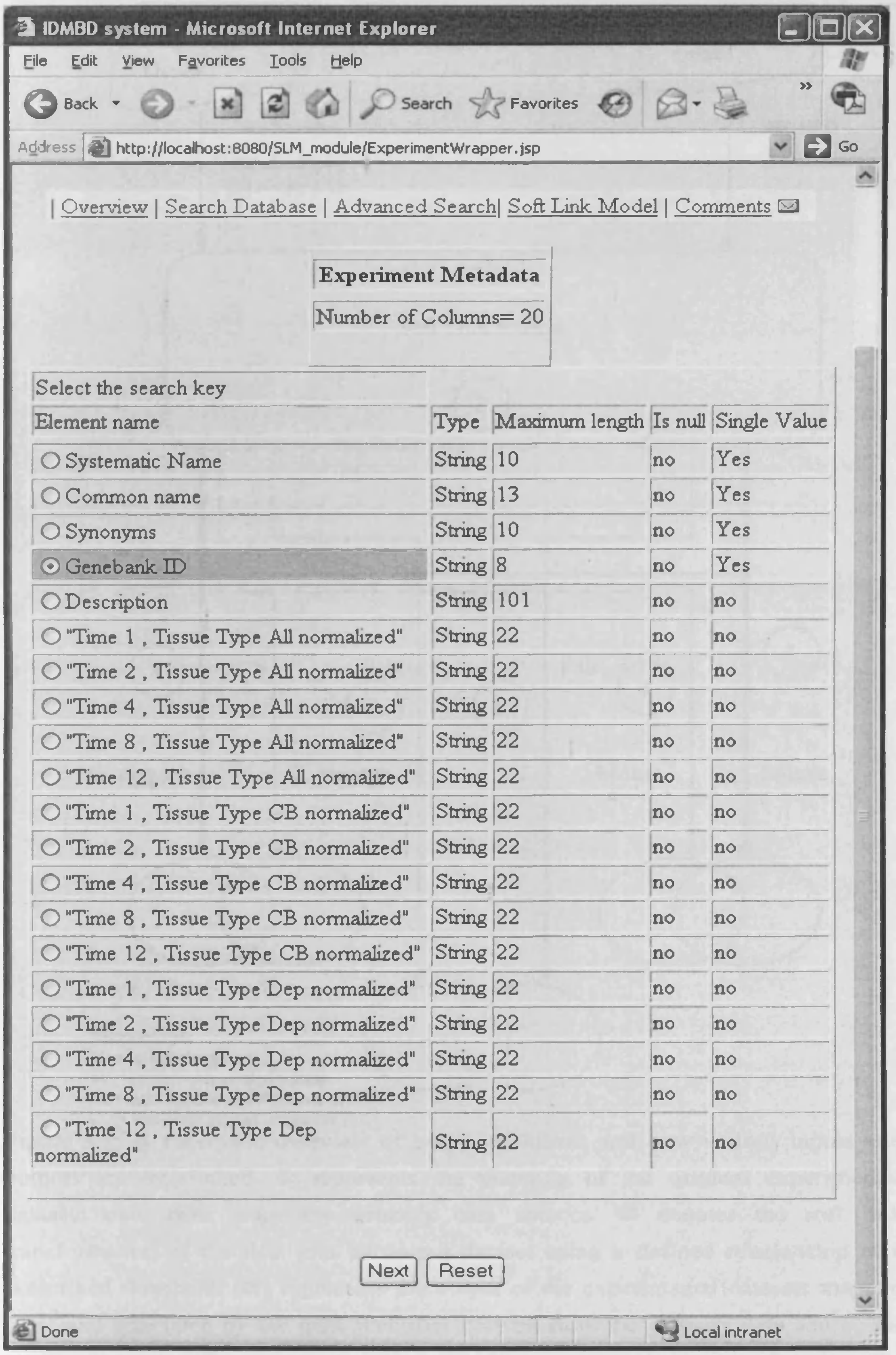
8.4.4 Defining genes conserved between species using specific functions and thresholds

Calculation of the intersection between species-specific identifiers is generated by converting the experimental identifiers or by transforming a complementary list from a second organism using a defined transformation function under a defined threshold. This intersection represents a group of genes conserved across species under the criteria defined by the transformation function and threshold.

8.4.5 Comparison and validation

To determine the impact of different transformation functions and thresholds we evaluated the intersection using various transformations. The aim was to enable us to identify and provide a biological explanation for the optimal threshold for each transformation and the elements that re-occur independent of the transformation function. The biological significance of the transformation process was calculated, for the mouse genes, by calculating the enrichment of specific biological processes and pathways against the processes/pathways represented by the large original list.

All *in silico* experiments were conducted using a platform equipped with an Intel Pentium 4 processor working at 2.80 GHz with 1 GByte of RAM, running Microsoft Windows, Sun Java Development kit 1.4 and Apache server 2.0.48. In the following sections, we present some of the more significant results from these experiments.

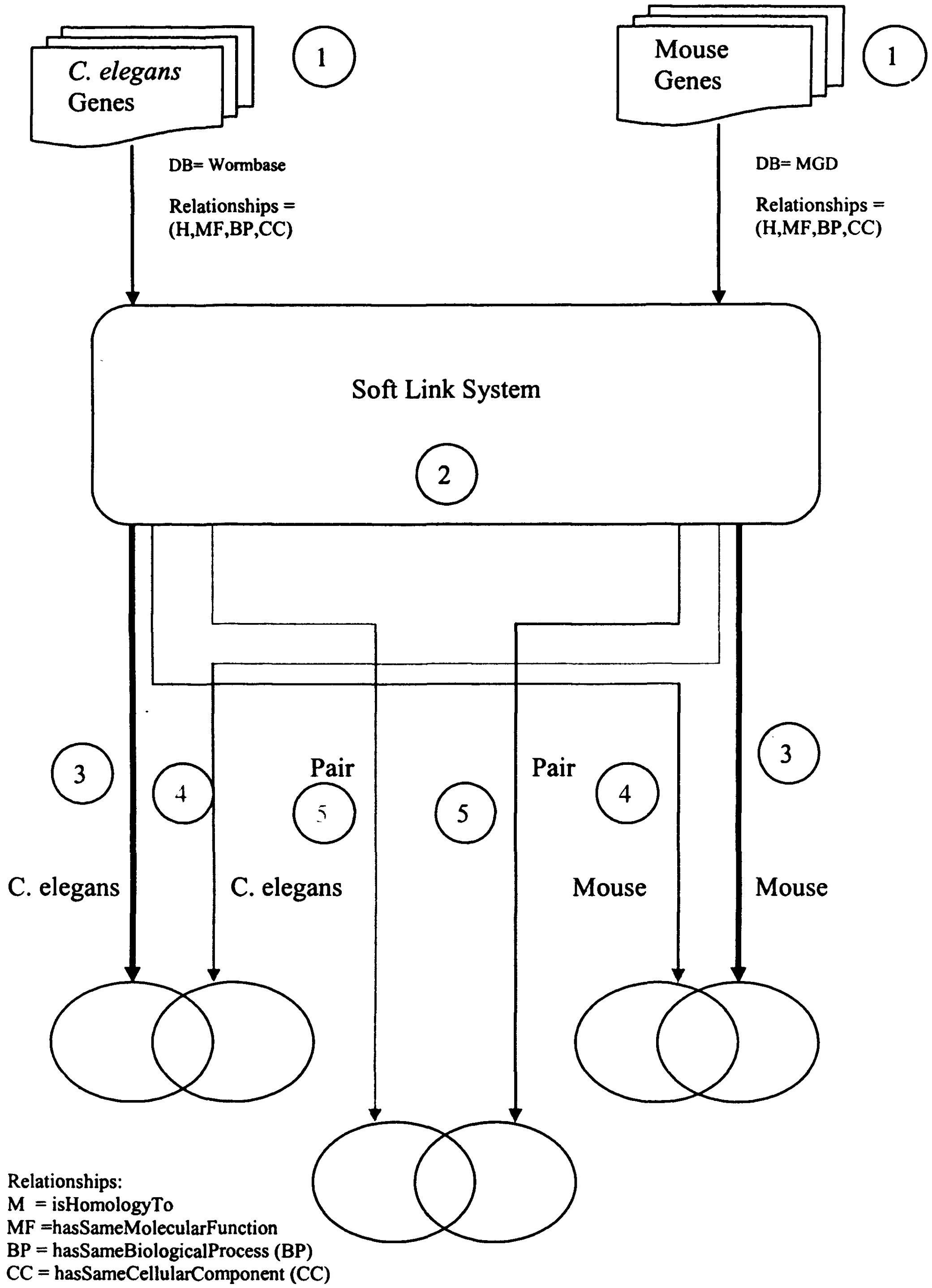

IDMBD system - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:8080/SLM_module/ExperimentWrapper.jsp
Go
| Overview | Search Database | Advanced Search| Soft Link Model | Comments
Next
Reset
Done
Local intranet

Experiment Metadata

Number of Columns= 20

Select the search key				
Element name	Type	Maximum length	Is null	Single Value
○ Systematic Name	String	10	no	Yes
○ Common name	String	13	no	Yes
○ Synonyms	String	10	no	Yes
● Genebank ID	String	8	no	Yes
○ Description	String	101	no	no
○ "Time 1 , Tissue Type All normalized"	String	22	no	no
○ "Time 2 , Tissue Type All normalized"	String	22	no	no
○ "Time 4 , Tissue Type All normalized"	String	22	no	no
○ "Time 8 , Tissue Type All normalized"	String	22	no	no
○ "Time 12 , Tissue Type All normalized"	String	22	no	no
○ "Time 1 , Tissue Type CB normalized"	String	22	no	no
○ "Time 2 , Tissue Type CB normalized"	String	22	no	no
○ "Time 4 , Tissue Type CB normalized"	String	22	no	no
○ "Time 8 , Tissue Type CB normalized"	String	22	no	no
○ "Time 12 , Tissue Type CB normalized"	String	22	no	no
○ "Time 1 , Tissue Type Dep normalized"	String	22	no	no
○ "Time 2 , Tissue Type Dep normalized"	String	22	no	no
○ "Time 4 , Tissue Type Dep normalized"	String	22	no	no
○ "Time 8 , Tissue Type Dep normalized"	String	22	no	no
○ "Time 12 , Tissue Type Dep normalized"	String	22	no	no

Figure 8.1: Screen snapshot shows the extracted metadata from the experimental datasets. The recommended linkage key is highlighted



Relationships:
M = isHomologyTo
MF =hasSameMolecularFunction
BP = hasSameBiologicalProcess (BP)
CC = hasSameCellularComponent (CC)

Figure 8.2: A schematic overview of query workflow, and how various inputs and outputs are interlinked. ① represents the mapping of the original experimental datasets onto their respective primary data sources. ② denotes the soft link transformation of the data into an output dataset using a defined relationship at a prescribed threshold. (③) represents the output of the experimental datasets mapped onto, and annotated by the gene identifier derived from the primary data source for source species. (④) represents the gene lists generated by transforming the experimental data using a defined linkage and threshold onto the gene identifier of a second organism. (⑤) represents intersections of the output list generated using various transformations

8.5 Results from SLM Analysis

In this section, we introduce the significant results obtained using the IDMBD system.

8.5.1 Orthological and Ontological Data Transformation

Data extracted from the two selected wet laboratory experiments (available as tab delineated flat files) were presented to the SLM system. Their metadata data were extracted (see Chapter 6) and mapped onto specific-species identifiers using the GenBank accession number as the linkage key. The Relationship Knowledge Base (RKB, Chapter 4) was then used to transform the datasets to lists of genes from the counterpart organism (i.e., transforming *C. elegans* genes onto mouse and vice versa). This process was performed using variable relationships and thresholds (see Chapter 5).

The mouse ortholog of the age-responsive *C. elegans* genes and *C. elegans* ortholog of the age-responsive mouse genes were determined using a BLAST transformation function. The number of orthologs identified was calculated under various levels of relatedness defined by varying the threshold for the probability of the sequence match occurring at random (i.e., the greater the probability, the lower the relatedness of the sequence) [120, 121]. This could be calculated only for probabilities <1E-1 (abbreviated to E-1) due to a threshold defined within the creation of the original RKB. In addition, ontological transformation of the two datasets onto specific-species identifiers for the complementary species was performed using the relationships of molecular function (MF), biological process (BP) and cellular components (CC).

The intersection between the original datasets, mapped onto their own specific-species identifiers, with genes representing complementary data from the other organism was also calculated. These measurements provide insight into the inter-species conservation of genes under a single transformation. Analysis considering a further intersection of genes transformed using multiple relationships, orthology and ontology, provides a gene list representing conservation of form (sequence) and function.

Table 8.2 shows the number of homology pairs between the two datasets at different thresholds.

Table 8.3 shows the number of MF, BP and CC pairs between the two datasets at different thresholds. The intersection between a Homology pair and Molecular Function (MF), and between a Biological Process (BP) and Cellular Component (CC) are 3278, 1814 and 13714 respectively.

Table 8.4 shows the intersection between a homology pair and a molecular function pair to homology pair at different thresholds. (Number of homology-pair $\cap$ number of similar-MF pair).

Table 8.5 shows the ratio of intersection between a homology pair and a Molecular Function pair to a homology pair. (Number of homology-pair $\cap$ number of similar-MF pair) / (Number of homology-pair).

Table 8.6 shows the number of GO-terms responsible for aging and growth from the datasets obtained, whereas Table 8.7 shows the ratio of GO terms responsible for aging and growth to total biological process GO across the two datasets.

Homology							
Threshold	0	E-70	E-40	E-30	E-20	E-10	E-1
HM	21	106	224	300	443	862	2214

Table 8.2: Number of Intersecting homolog pairs between two datasets at different thresholds

Relationships	Number of pairs
Molecular function (MF)	3278
Biological process (BP)	1814
Cellular component (CC)	13714

Table 8.3: Number of Intersection of MF, BP and CC pairs between two datasets

Threshold	0	E-70	E-40	E-30	E-20	E-10	E-1
HM X MF	0	6	9	12	17	29	38
HM X BP	0	14	33	45	56	102	189
HM X CC	0	10	15	16	23	34	67
HM	21	106	224	300	443	862	2214

Table 8.4: Intersection between homology pair and MF, BP and CC

Threshold	E-70	E-40	E-30	E-20	E-10	E-1
1.MF	0.056604	0.040179	0.04	0.038375	0.033643	0.017164
2.BP	0.075472	0.084821	0.07	0.049661	0.034803	0.01897
3.CC	0.09434	0.066964	0.053333	0.051919	0.039443	0.030262

Table 8.5: Fraction of MF, BP and CC to homology across mouse and C. elegans. Mapping mouse age-related genes onto C. elegans components using different relationships and thresholds. These figures are calculated by: 1.MF= (HM X MF)/HM, 2.BP= (HM X BP)/HM, and 3.CC=((HM X CC)/HM

Threshold	E-70	E-40	E-30	E-20	E-10	E-1
Number of aging & growth with MF relationship	3	4	5	8	13	20
Number of aging & growth with BP relationship	3	4	4	4	7	10
Number of aging & growth with CC relationship	3	3	4	7	8	11
Total biological process	14	33	45	56	102	189

Table 8.6: The number of genes with GO-terms related to aging and growth

Threshold	E-70	E-40	E-30	E-20	E-10	E-1
MF	0.214286	0.121212	0.111111	0.142857	0.127451	0.10582
MP	0.214286	0.090909	0.088889	0.125	0.078431	0.058201
CC	0.214286	0.121212	0.088889	0.071429	0.068627	0.05291

Table 8.7: The ratio of genes with GO terms related to aging and growth to the total with conserved ontological classification across two datasets

8.5.2 Determining the optimal threshold for cross-species orthology relationship

C. elegans orthologs were calculated for the cohort of age-regulated mouse genes at variable levels of relatedness and the intersection of this group was calculated with a complementary transformation of the mouse genes using ontological categorization, through direct or parent-child association. This provided us with a profile (Figure 8.3) that described the relationship between protein sequence conservation (as expressed by the homology score) and maintenance of the biological role. A clear optimum, for both Cellular Component (CC) and Molecular Function (MF) can be identified, where the expected probability of a match is between E-70 and E-40. This represents a small group of highly conserved genes displaying significance in their area of biological function. This proportion of genes with matching MF and CC ontologies drops sharply until it reaches a plateau, 4% for MF and 6% for CC, between E-40 and E-10. This shows that decreasing the stringency of orthology identification over a significant range does not reduce the proportion of genes with matching ontologies. This implies that the increased number of orthologs identified is not increasing the proportion of random or non-specific matches. It is evident that this profile can be used to identify the optimal threshold at which to perform cross-species data mining. Approaches employing high or low cut-offs either discard useful data or include substantiale noise.

Intriguingly, the profile for the proportional intersection for Biological Process (BP) terms is different and does not show the biphasic properties of MF and CC. Instead, a smooth curve is seen with a broad optimum at ~E-40. The percentage of intersection falls off smoothly until it reaches that attributable to random matches at E-1. This unique profile may be a property of the highly diverse nature of the biological processes between these two species or due to the heterogeneity of gene annotation by the communities. The data generated suggest that functional interpretation of cross-species using an orthology model must be informed by the specific inter-species relationship between orthology and function.

Threshold	E-70	E-40	E-30	E-20	E-10	E-1
MF	0.056604	0.040179	0.04	0.038375	0.033643	0.017164
BP	0.075472	0.084821	0.07	0.049661	0.034803	0.01897
CC	0.09434	0.066964	0.053333	0.051919	0.039443	0.030262

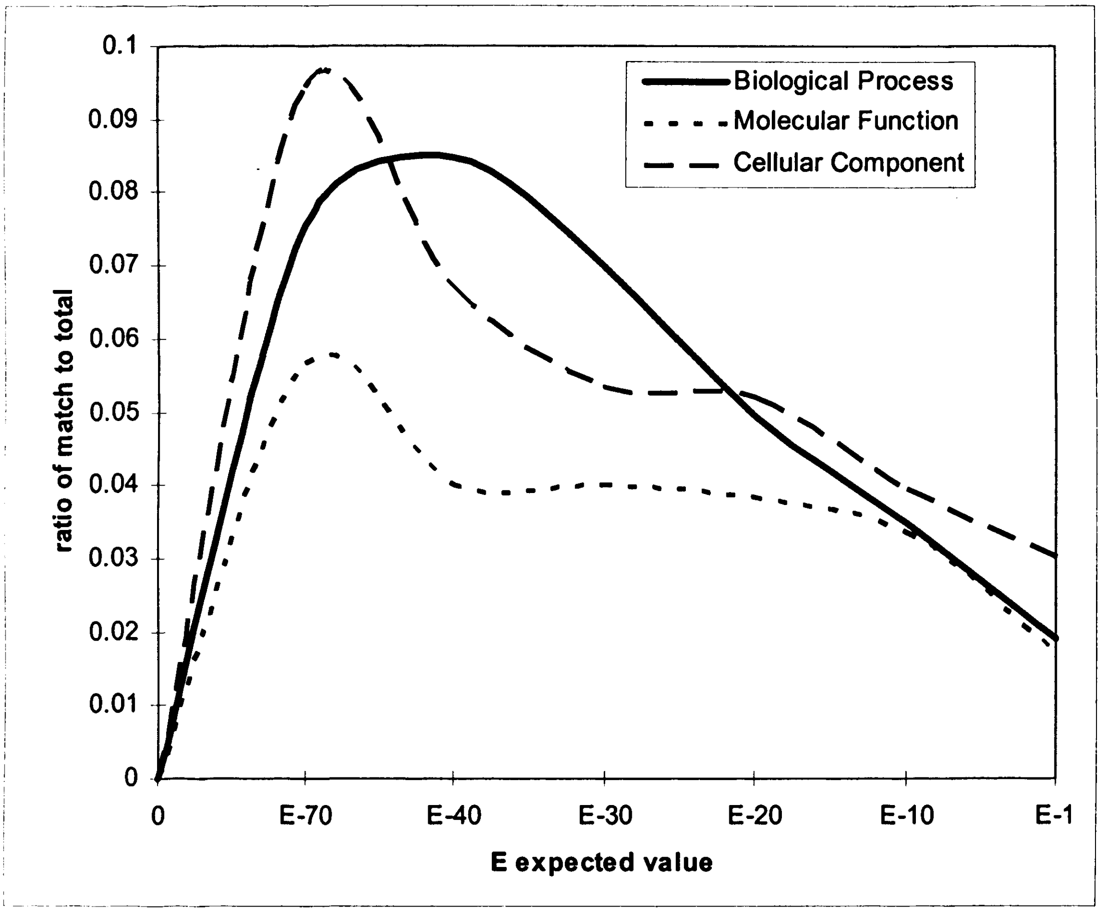


Figure 8.3: The profile of the relationship between protein sequence conservation (as expressed by homology score) and maintenance of the biological role. A clear optimum, for both Cellular Component (CC) and Molecular Function (MF), could be identified where the expected probability of match is between E-70 and E-60.

8.5.3 Investigating the consequence of variable thresholds when defining the intersection of evolutionary and functional conservation

The exemplar experimental sets were designed to investigate the transcript responses to aging; therefore, it was important to establish whether those genes experimentally determined as aging-related showed an established ontology relating to "aging" or "growth" that was conserved across the species boundaries. We explored this overlap between the homolog of the cohort of mouse genes displaying up-regulation in response to age with an ontological category in both mouse and *C. elegans* defined as "age" and "growth". This intersection was determined for the three ontological classes BP, MF and CC using a wide range of orthology thresholds. These data displayed profiles similar to those determined for the global conservation of all ontological categories determined previously. There is a clear maximum on the proportional representation at E-70 with a secondary feature peak at E-20 (see Figure 8.4). This indicates the presence of a group of "aging or growth" genes displaying high overall conservation with a smaller number of genes, which exhibit less conservation; this latter group may arise from moderate overall conservation or may be attributed to the conservation of key functional regions. This former observation is consistent with the recognised functional architecture of proteins that exploits common and flexible secondary structural motifs to support key functional residues, whereas the latter explanation reflects the evolutionary attribute of functional domains being used within variable protein architectures. What is intriguing is that the proportion of genes within this group displaying conserved ontology "aging" or "growth" classification is 10 times higher than that observed for all ontological categories. This may suggest that the genes involved in aging and growth are much more highly conserved across the wide evolutionary gap between mouse and *C. elegans*. It is clear from the data that by exploiting the variable threshold, we can define either a cross species mapping that is extremely conservative, identifying an orthology group that has a maximum probability of sharing function whilst the selection of a lower threshold will permit the maximum return of related genes and still minimise the noise generated from random matches.

Threshold	E-70	E-40	E-30	E-20	E-10	E-1
MF	0.214286	0.121212	0.111111	0.142857	0.127451	0.10582
BP	0.214286	0.121212	0.088889	0.071429	0.068627	0.05291
CC	0.214286	0.090909	0.088889	0.125	0.078431	0.058201

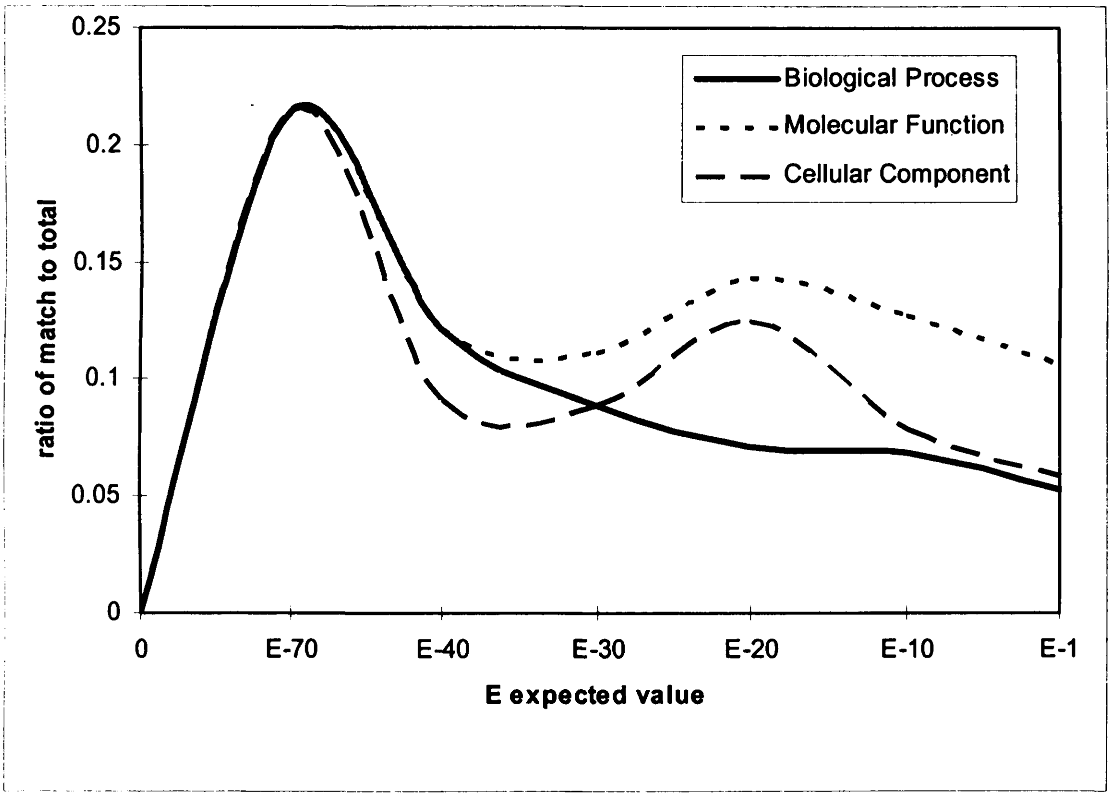


Figure 8.4: A graph exploring the overlap between the homolog of the cohort of mouse genes displaying up-regulation in response to age with an ontological category in both mouse and C. elegans defined as "age" and "growth". This intersection was determined for the three ontological classes BP, MF and CC using a wide range of orthology thresholds.

8.5.4 Functional enrichment through cross-experimental comparison

In the following section, we discuss the outcome of the experiments to show functional enrichment through cross-experimental comparison across species. We use the DAVID system to show the enrichments.

The SLM implementation enabled us to compare the molecular responses detected in an aging experiment performed in mouse and the model nematode *C. elegans*. It is illustrative of a major challenge for genomics studies that these experiments implicate substantive numbers of genes within the aging process; our analysis yielded 500 mouse and >3500 *C. elegans* genes, which increased during aging. It is impractical to investigate this plethora of possible targets experimentally. Therefore, techniques that can refine the lists to those targets that are central to the biological parameter under investigation are essential to the investigators to enable them to focus on realistic subsequent wet experimentation. In theory, the ability to identify elements that respond in the same manner across species should achieve the goal of identifying evolutionarily and functionally conserved elements.

In order to characterise the refinement process under varied methods and thresholds of linkage we analysed the SLM output in relation to the 500 aging-responsive mouse genes, since this species has a higher degree of annotation than has *C. elegans*. This initial cohort was used as a "background" population and the functional enrichment of the intra-species conserved sub-groups calculated [63, 106]. The use of orthology to map the *C. elegans* aging-related genes onto their mouse counterparts (MGI ID) allowed the inter-section of these two groups to be calculated. Using an orthology threshold of E-10 (defined by the BLAST probability score) an intersection of 104 unique gene objects could be identified whilst an increased stringency of E-70 yielded only 60 gene objects. Significant and subtly different functional enrichment was observed in both groups (see Figures 8.5 & 8.6). The lower stringency orthology displayed enrichment in functional annotation categories relating to transcriptional control, replication and chromatin amongst others (see Figure 8.5). Increasing the threshold to consider only those genes that exhibit extremely high homology (E-70) gives functional groups related to nucleotide binding, replication, cell cycle and protein/cellular metabolism (see Figure 8.6). These sets are not exclusive, but the enrichment scores for each group are subtly different indicating the impact of altering the threshold.

When mapping the *C. elegans* aging related genes onto their mouse counterparts using a "molecular function" ontology the result was a large, highly repetitive list that yielded a non-redundant set of 289 mouse genes with Ensembl IDs. When this list was used for enrichment analysis, it yielded far weaker enrichment scores, but the functional groups generated were associated with the mechanism of regulation as may be expected from a mapping molecular function. These groups included those genes involved in phosphylation (kinases), DNA modification and the regulation of cell processes (Figure 8.7).

The intersection between lists generated by orthology and ontological linkage provided a focused subset of genes, 16 and 6 under orthology thresholds of E-10 and E-70 respectively, when analysed for conserved molecular function. Analysis of the less stringent group identified overrepresentation of members of pathways including cell cycle and focal adhesion whilst the higher stringency group indicated only a bias for elements involved in the cell cycle process. These are processes known to have a close link to aging and cell maintenance and therefore the specific genes identified by this process may potentially form high priority targets for further investigation.

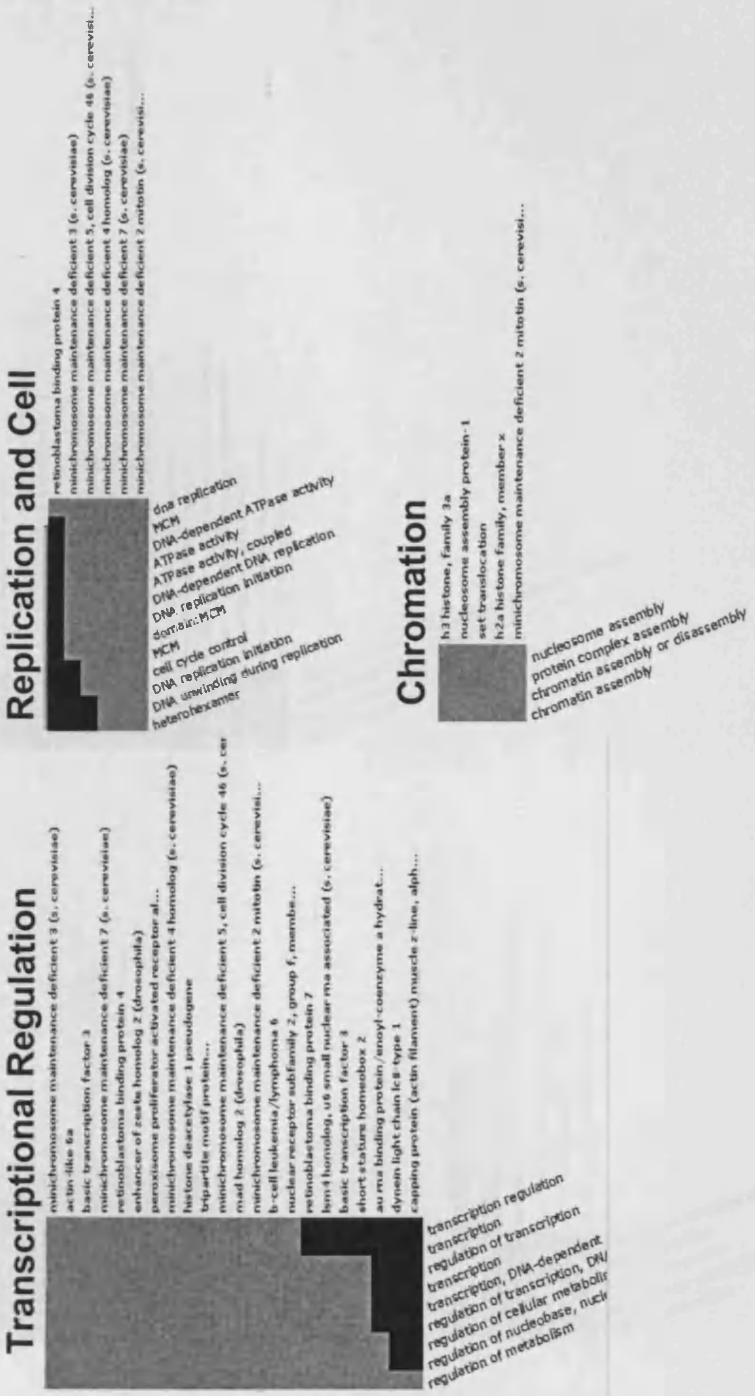


Figure 8.5: David Functional annotation clustering using classification stringency “high” employing a gene list derived using the intersection provide full description MC-10 Pair. The use of orthology to map the C. elegans aging related genes onto their mouse counterparts (MGI ID) allowed for the inter-section of these two groups to be calculated. Using an orthology threshold of E-10 an intersection of 104 unique gene objects could be identified. Significant and subtly different functional enrichment was observed in the group. The lower stringency orthology displayed enrichment in functional annotation categories relating to trasnactional control, replication and chromatin amongst others.

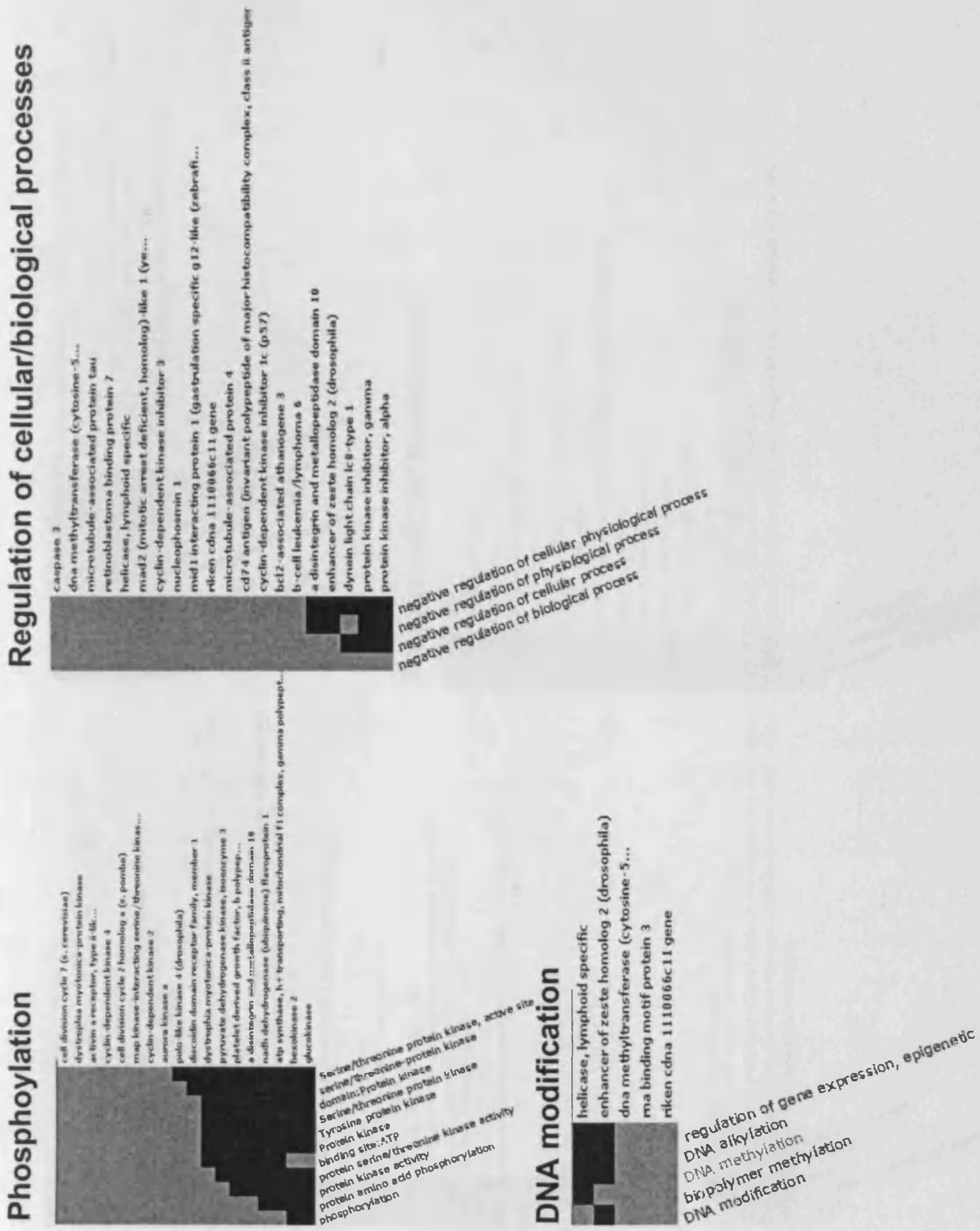


Figure 8.6: David Functional annotation clustering using classification stringency "high" employing a gene list derived using the intersection provide full description MC-70 Pair. The use of orthology to map the C. elegans aging related genes onto their mouse counterparts (MGI ID) allowed for the inter-section of these two groups to be calculated. Using an orthology threshold of E-70 it yielded only 60 gene objects. Increasing the threshold to consider only those genes which exhibit extremely high homology (E-70) gives functional groups related to nucleotide binding, replication, cell cycle and protein/cellular metabolism.

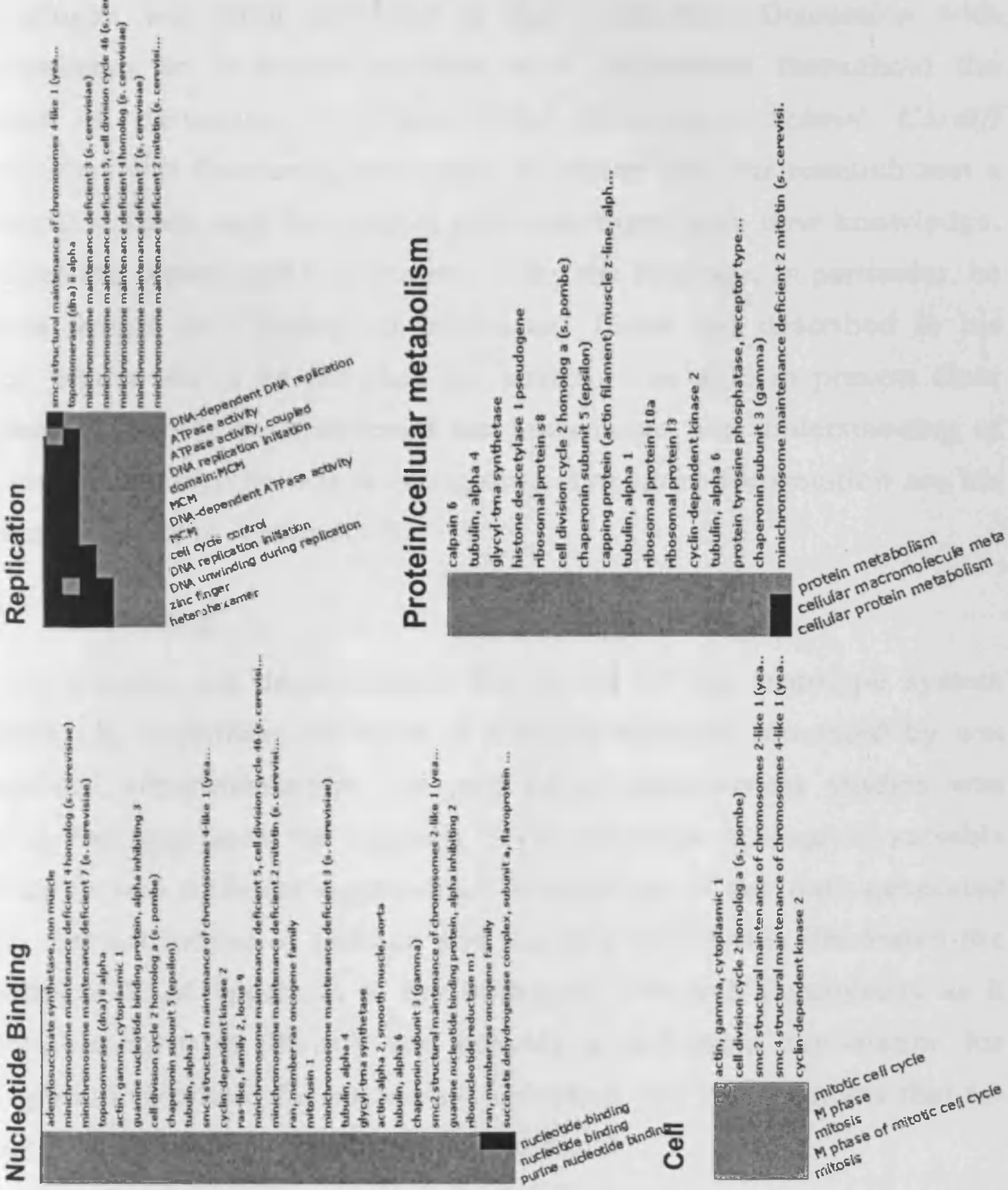


Figure 8.7: David Functional annotation clustering using classification stringency "high" "employing a gene list derived using the intersection provide full description MC-MF Pair. When mapping the C. elegans aging related genes onto their mouse counterparts using a "molecular function" ontology the result was a large highly repetitive list which yielded a non-redundant set of 289 mouse genes with Ensembl IDs. When this list was used for enrichment analysis it yielded far weaker enrichment scores but the functional groups generated were associated with mechanism of regulation as may be expected from a mapping molecular function. These groups included those genes involved in phosphylation (kinases), DNA modification and the regulation of cell processes.

8.6 Biologist evaluation

A biologist was fully involved in this evaluation. Discussion with professionals in biological science was undertaken throughout the project. In particular, Dr. Peter Kille (*Bioscience School, Cardiff University*) was frequently consulted to ensure that our research met a biologist's needs and the system provides them with new knowledge. He used the system and was impressed by the findings. In particular, he gained insight into biological problems. These are described in his letter, which shows he felt that the system was able to present clear information which he broadened his knowledge and understanding of the area of biology he was investigating. For more information see his evaluation letter in Appendix E

8.7 Summary

In this chapter, we demonstrated the utility of the prototype system IDMBD, by exploiting the tools to analyse datasets generated by wet laboratory experimentation. A pair of complementary studies was chosen that represent the analysis of an identical biological variable studied in two different organisms. Evaluation of the data generated using distinct modes of linkage and variable thresholds illustrated the benefits of SLM approach to the biological research community as it enable biologists to identify and provide a biological explanation for the optimal threshold for each transformation and the elements that re-occur independent of transformation function.

Chapter 9

Evaluation

9.1 Synopsis

We implemented a version of IDMBD as an illustration of concept prototype, which discovers relationships in heterogeneous bioinformatics data sources based on biological relationships between biological objects across species. These were then used to integrate the data sources. In this chapter, we evaluate our framework and objectives as well as considering the key issues about IDMBD.

9.2 Introduction

Comparative genomics is the analysis and comparison of genomes from different species. Its aims are to gain a better understanding of how species evolved and to determine the function of genes for which no experimental evidence currently exists. Comparative Genomics provides a powerful set of tools for leveraging information across species. For example, the functions of the human genes have been discovered by examining their counterparts in simpler model organisms such as mouse. Comparative analysis is hypothesis driven and thus a biologist requires the ability to ask "what if" questions to test theories on the whole genome, such as its organization, structure and evolution. Usually, genome researchers look at many different features when comparing genomes such as sequence similarity, gene location, and highly conserved regions in the genetic sequences.

By integrating functional and sequence data across species, we are able to annotate the genome of one species using known functional data about another. Thus, comparative genomics provides evidence using close evolutionary relationships between gene families.

Comparative genomics involves the use of various bioinformatics tools such as sequence-similarity tools and GO-term similarity. These tools have different interfaces and often involve transforming the output from one tool into a format suitable as input to another tool. This means that a researcher has to do manual tasks, such as cutting and pasting data or identifying the tool that will transform the data appropriately. This is an error prone process and is time consuming. Thus, what is required is a system that allows biologists to take the results of one analysis and use them as the basis for conducting further downstream analysis in a manageable, flexible, quick, accurate and efficient way by inputting the data to subsequent tools.

Bearing in mind that the aim is to develop a system that facilitates the determination of functional annotation and analysis of large sets of genes, IDMBD aims at automating the process of comparative genomics and data integration as far as possible.

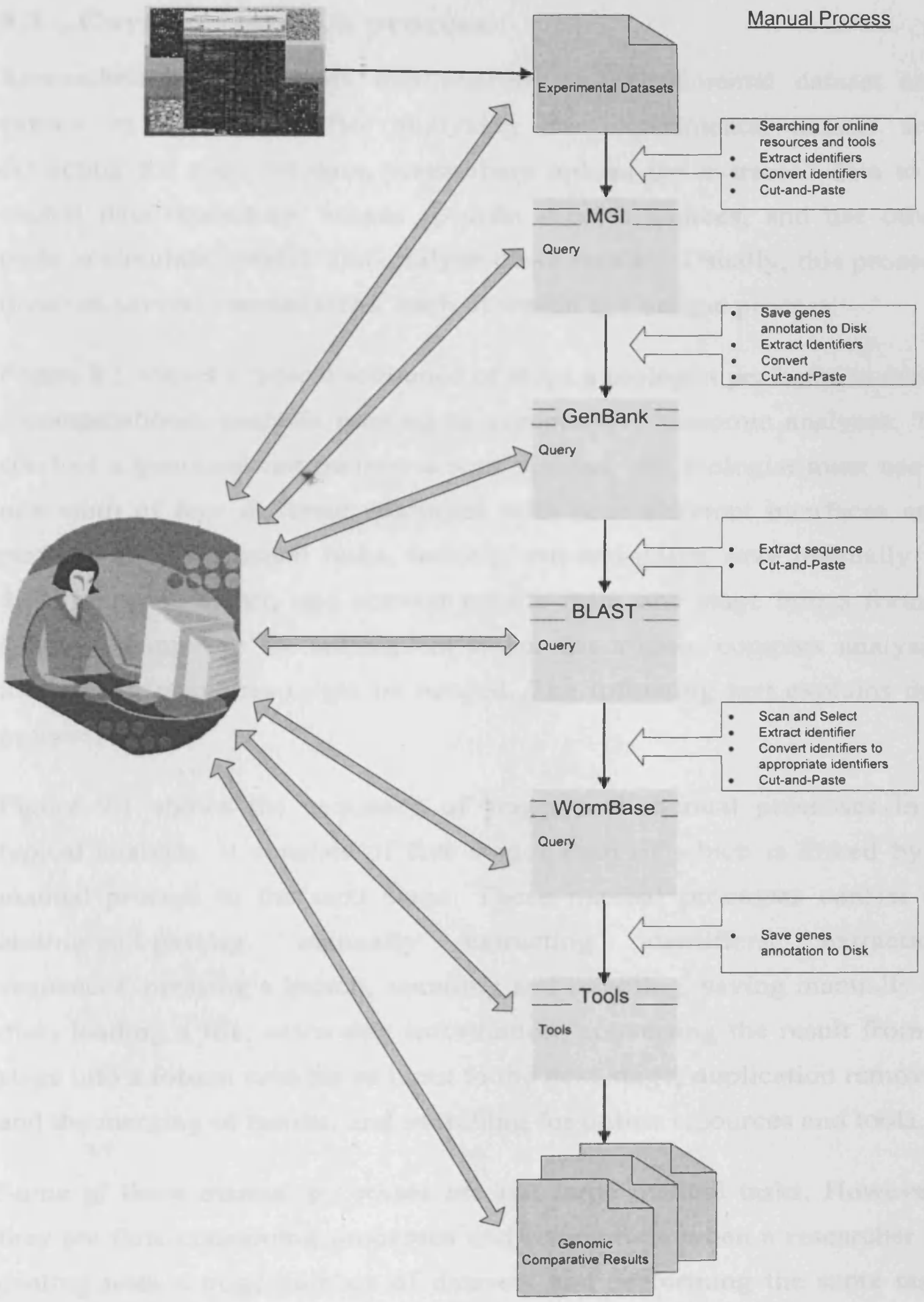


Figure 9.1: Typical sequence of steps a biologist performs to drive a series of computational analyses relating to comparative genomic analyses

9.3 Current research process

Researchers develop tools that analyze an experimental dataset and extract its metadata. After analyzing the experimental dataset and extracting the required data, researchers upload the extracted data to a central data repository, access specific species sources, and use other tools to simulate, model, and analyze these results. Usually, this process involves several manual steps, each of which is a unique process.

Figure 9.1 shows a typical sequence of steps a biologist performs to drive a computational analysis relating to comparative genomic analyses. To conduct a genomic comparison across species, the biologist must use a minimum of four different resources with four different interfaces and perform several manual tasks, namely, cut-and-paste, save manually to disk, scan and select, and convert results from one stage into a format suitable as input to the subsequent stage. For a more complex analysis, many other resources might be needed. The following text explains this process.

Figure 9.1 shows the sequence of stages and manual processes in a typical analysis. It consists of five stages each of which is linked by a manual process to the next stage. These manual processes consist of cutting-and-pasting, manually extracting identifiers, extracting sequences, pressing a button, scanning and selecting, saving manually to disk, loading a file, extracting information, converting the result from a stage into a format suitable as input to the next stage, duplication removal and the merging of results, and searching for online resources and tools.

Some of these manual processes are not large manual tasks. However, they are time-consuming processes and error-prone when a researcher is dealing with a huge number of datasets and performing the same task hundreds of times. For example, cut-and-paste is not a large manual task but it is still prone to error, while the scan and select is a much larger process since a researcher has to scan through output and decide on parameters to obtain results of interest. The mapping of an accession number to a specific species identifier is not an easy task and may need the use of other tools. Converting the result from a tool into a format suitable as input to a subsequent tool requires time and effort.

During an experiment, a biologist will usually perform a series of computational analyses on their data (see Figure 9.1), as follows:

1) When the experimental datasets are in a file, the dataset is parsed to extract manually the up-regulated gene identifiers and save them. The biologist then has to map identifiers to species-specific identifiers, for example, an MGI identifier. He/she may need a tool to convert the specific species identifiers to standard accession numbers and vice versa. The biologist then uses his/her past experience or searches online for resources and tools related to the species of interest.
2) When a web-based resource offering species-specific genomic data is identified, the biologist uses the interface provided to fill up the form with identifiers or gene names and query the source to retrieve gene annotations connected to the gene list.
3) Upon the conversion of identifiers to appropriate accession numbers, the biologist accesses, browses and queries sequence databanks, such as NCBI Entrez. Using the interface provided by the tool, the biologist pastes accession numbers, sets up his/her parameters, and submits a query. When the sequence is retrieved, he/she extracts the corresponding sequences, which can be saved to disk.
4) The researcher can perform a similarity search using a public BLAST resource with the sequences obtained in the previous step, and filter the results in some way to find similar genes in related species. He/she then saves the hits, extracts results and looks manually for sequences from the related species of interest with required parameters. Then he/she saves the results and manually extracts accession numbers of sequences above a specified threshold value using either an identity percentage or an E-value.

5) The biologist maps accession numbers to species-specific identifiers and identifies a web site offering related species-specific genomic data. The interface is used to fill up the form with identifiers and query the source to retrieve the gene annotations connected to the gene list identified in previous step.
6) The biologist uses other tools to analyse and map the results obtained from steps 2, 3 and 5 and to gather results from different species, remove duplication, and map the gene annotations obtained from different species to predict gene function or other features of the experimental genes with similar or related genes having known functions. The result of analysis and the comparatives are then saved to disk.

This process is usually repeated for each analysis undertaken and for each new experiment and new approach of linkage (homology, orthology, or ortholog). Thus, if a biologist wants to find orthology genes from other species to identify evolutionary changes, the steps in this process have to be undertaken again. This also occurs if a comparison uses the GO terms between different genes to predict gene functions, when steps 4 to 7 have to be repeated. This is a well known problem; for instance, Troup [192] stated that to drive the experimental process, the biologist is hampered by at least four distinct problems:

1- Discovery of Bioinformatics resources

Biologists have to browse, search, and access multiple data sources and bioinformatics tools before discovering an appropriate solution that can be used to create and evaluate a new biological hypothesis. To drive this experimental process and perform the analyses only on datasets of interest, involves the following steps:

- Searching the internet for primary sources and bioinformatics tools.
- Selecting relevant bioinformatics data sources and tools.

- Accessing the selected data sources.
- Retrieving the data or using data analysis tools.

A considerable amount of time and energy is needed to find relevant data sources and tools and access them. It also involves many manual transfers, which can be prone to mistakes

2- Data format conversion

As the biological data sources and bioinformatics tools have been developed over time by different communities, biological data are stored and distributed in a wide variety of formats, which are often not consistent or interchangeable. Usually, a researcher takes the results of one analysis of data as the basis for conducting further downstream analyses in a manageable and efficient way. With a diverse range of file formats and representations of bioinformatics data, it has become an increasingly difficult task for a researcher to deal with the different formats and analysis tools. Thus, a researcher wishing to perform multiple analyses of data by feeding the results of one program into another continually encounters the issue of converting data from one format into another. This is often a very difficult and time-consuming process, which is error-prone.

3- Manual transfer of data

Normally, a biologist takes the results of one analysis as the basis for conducting further downstream analyses. Thus, it is necessary to move data between very different systems with different representation formats. Traditional ways of accomplishing this transformation include the use of copy-and-paste, menu-driven interfaces, and a command line. These mechanisms are adequate for small tasks; however, they do not scale to large tasks, as they involve performing the same task hundreds of times. Thus, these manual mechanisms make the task tedious and time-consuming as well as error-prone during the transfer of data between systems.

4- Understanding how to use the various tools on a variety of platforms

The massive increase in the number of bioinformatics tools that often run on different platforms means it is not an easy or practical task for a biologist to learn about each individual tool and how to integrate it with other tools. To gain benefit from the available tools there is a need to understand and manage different tool platforms. Thus, learning and managing these tools is both time-consuming and difficult and needs expertise in the tool for its effective use.

Thus, biologists spend a lot of time and effort dealing with data sources and tools. A previous study [4] claimed a biologist spends more than 50% of the analysis time on tasks related to manipulating data from incompatible data sources and using tools to change them to the required new formats.

We have shown the process consists of stages with manual processes between stages, all of which take time. This is the normal way that biologists conduct this type of research. In recent years automated approaches have started to appear which automate some of the manual processes. Most notable of these are systems based on a workflow approach, for example myGrid [183]. Our approach is an alternative way of automating the stages to a workflow.

Workflows (for example Taverna in myGrid) automate some of the process in the flow shown in Figure 9.1. However, workflows can themselves become complex. As they may involve several stages, each of which is time-consuming, difficult and needs expertise to successfully undertake the stages. In order to create an appropriate workflow, the biologist has to put in place the following stages [109]:

- Service discovery: the biologist has to identify services that perform the task needed for the experiment. Thus the biologist has to construct a new workflow each time and often change the linkage type. However, services can be difficult to find because they are poorly described and changing linkage type is not always a straightforward process as the description can be vague and menimalistic.

- Service Gluing: the biologist has to identify how services are compatible and fit together. However, joining services together into a workflow is frequently problematic, as the inputs and outputs are not directly compatible. Consequently, many Shim services [108] are needed to align inputs and outputs in a workflow and enable services to interoperate.

- Service invocation: the biologist need to know how to invoke the services, what data and parameters are needed.

As a result, a minority of biologists are likely to construct workflows [67]. An additional problem may b that every time the analysis changes, the data may have to be re-transferred from the source, a time consuming operation.

On the other hand, with IDMBD, a biologist has only to specify the experimental datasets' file, the relationships type, the relationship closeness and the information wanted. Moreover, the biologist can repeat the experimentation with different relationship and relationship closeness measures, easily and quickly without the need to construct a new environment as in workflows or re-transferred data.

If an appropriate workflow is available it may be easy to re-use it thus saving the time of creating from scratch this element but this is only the case if the same analysis is required. If the biologist needs to repeat the same analysis many times then the workflow will be ideal. Also if the biologist has the skills to build the stage linkages when new stages are inserted into a workflow then the workflow approach will meet his/her requirements. Generic workflow systems, such as Taverna, have been used for some time and are often part of much wider tool set which has a variety of sophisticated display and analysis tools which can be utilised in sophisticated analysis and presentation of results, e.g. graphic analysis. This situation is not present in the IDMBD environment.

9.4 The IDMBD approach

To alleviate some of these problems, we developed the IDMBD system (Figure 9.2) to automate entirely the processes of Figure 9.1. Its user has only to specify the experimental datasets' file, the species name, the relationships type, the relationship closeness and the gene annotation wanted in order to use the system.

9.4.1 SLM

SLM is a novel approach to interoperation that is based on the use of biological relationships. A relationship that exists between biological objects is an important factor in linking bioinformatics sources as it can effect the integration of bioinformatics data sources. Unlike current integration strategies, which focus on using ontology-based or keyword-based linkage, we used relationship-based integration to integrate bioinformatics data sources. This is achieved in our framework by introducing the Soft Link Model and a relationship knowledge base (RKB), which is built and used by SLM.

SLM consists of concepts, relationships and degrees of linking. A concept is an entry in a database that represents a real-world entity. The SLM models the linkage between data sources in terms of concepts, properties and semantic relationships (see section 4.4).

The Relationship Knowledge Base (RKB) is a collection of relationship tables that hold Source_id, target_id, RelationshipType, and Relationship Closeness, which store semantic relationships between biological objects. These relationships between sources are exploited to combine annotation knowledge from different sources. RKB is used to link datasets with other public data sources. There is no need to perform a comparison between species during the run-time process since this is done as a separate task and stored in RKB. This saves time and effort as they can be used in several analyses.

We identified a gene-product concept in two sources. For homolog, ortholog, and paralog relationships, we chose sequence properties in both

As explained in section 5.5, the system performs the analysis as follows:

1. The biologist uploads the experimental datasets from delimited flat files via the user interface through a standard web browser; the system parses the datasets, extracts the metadata and converts the dataset into an appropriate format.
2. It detects the suitable linkage key based on the scoring table (Table 6.1 in section 6.3.1) and shows the metadata to the user who can confirm the recommended linkage key or choose a different key from this metadata.
3. The user then sets up a query and feeds the system with the species name to be used in the experiment, the relationship type to be used for linkage with the species, the relationship closeness and the required gene annotation to be retrieved.
4. When the mediator receives a user query and experimental dataset, it selects the primary species-specific source to answer the query. Upon selection of the source, it generates a retrieval query to specific species using accession numbers or identifiers.
5. The mediator invokes the wrapper of the selected source.
6. The selected wrapper connects to the data source by means of its standard API and submits the query to the data source.
7. On receiving results from the data source, the wrapper passes them to the mediator.
8. The mediator extracts the gene identifiers from the result set and then generates a new call to the Soft Link Model to retrieve all relationships associated with this Gene concept from other species. It sends the source name, concept, identifiers and user preference - the relationship that the user wants to use to link data sources and the relationship closeness cut-off.
9. The Soft Link module loads the SLM metadata and searches whether any relationships associated with the concept and data sources have been sent to it by the mediator. If a relationship specified by the user is found between the selected concept from the data source and concepts in other data sources, the Soft Link

module invokes the relationship wrapper, which opens a connection to the RKB and fetches instances that satisfy user preferences. Basically, it fetches related concepts, related data sources, and identifiers of related entries in the related source.

10. The Soft Link Adapter then responds to the mediator with a list of related identifiers and related source concepts.
11. When the mediator receives the response, it links to other species-specific sources via wrappers to retrieve all related genes from those sources that may have relationships with the target source.
12. The mediator recomposes the various responses and formats the final response to the user.

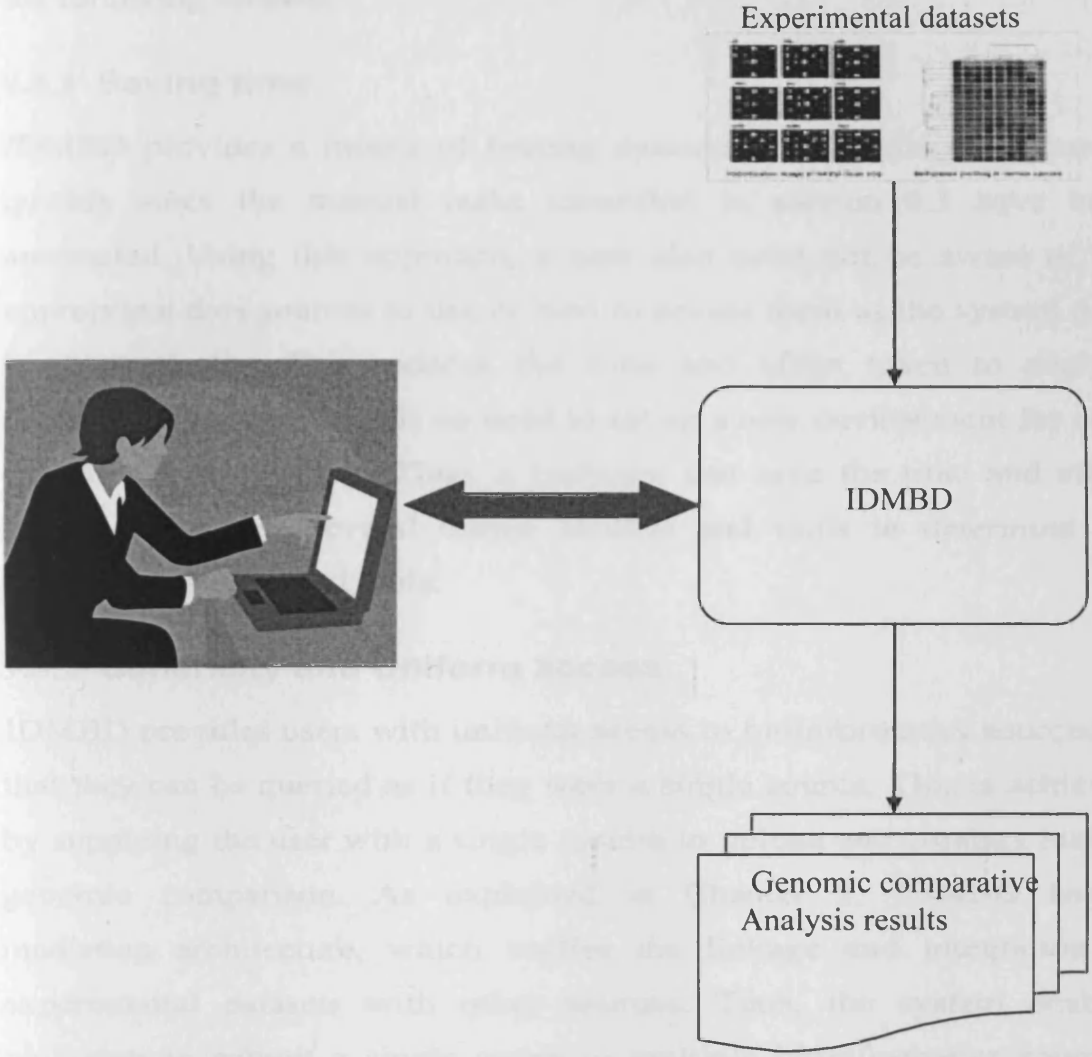


Figure 9.2: Sequence of steps a biologist performs using IDMBD to drive a series of computational analyses relating to comparative genomic analyses

9.5 IDMBD evaluation

We have conducted research in computer science and bioinformatics. In order to evaluate our system's potential, we need to test how much it improves the comparison process against the current approaches or manual execution of the desired tasks. Most importantly, we need to assess the effectiveness of the system as a tool to help biologists conduct this type of analysis in a fast, practical and easy way. The evaluation metrics will be time, genericity, intervention, transparency, flexibility, extensibility, heterogeneity and functionality; we consider these issues in the following section.

9.5.1 Saving time

IDMBD provides a means of linking datasets with public data sources quickly since the manual tasks identified in section 9.3 have been automated. Using this approach, a user also need not be aware of the appropriate data sources to use or how to access them as the system does it automatically. This reduces the time and effort taken to analyze datasets. Moreover, there is no need to set up a new environment for each experiment on the data. Thus, a biologist can save the time and effort needed to browse several online sources and tools to determine the appropriate source and tools.

9.5.2 Genericity and Uniform access

IDMBD provides users with uniform access to bioinformatics sources so that they can be queried as if they were a single source. This is achieved by supplying the user with a single system to upload and conduct his/her genomic comparison. As explained in Chapter 5, IDMBD has a mediation architecture, which unifies the linkage and integration of experimental datasets with other sources. Thus, the system enables biologists to submit a single query to multiple bioinformatics sources, and returns a unified set of results. This means a user does not need to spend time submitting the same query over and over again to many data sources. IDMBD is also generic in that it is not designed to answer a single query. Instead, it offers several alternative linkages for the integration of sources that can be used by the researcher without further work.

9.5.3 Reducing human interaction

Wherever possible, IDMBD system automates manual tasks to minimize human interaction. It permits the automated extraction of the experimental dataset's metadata, the analysis of its contents and integration with other bioinformatics sources to enrich annotation without intervention. As can be seen in Figure 9.2, there is a single user interaction with this approach while Figure 9.1 shows that there are many interactions. Thus, our system reduces the number of human interactions from seven interactions (see Figure 9.1) to one interaction (see Figure 9.2). The system automates the entire procedure without human intervention. Human intervention is required only to supply the experimental dataset file, decide on parameters, and make the decision to select the linkage key. There is thus a clear saving in human interaction. Sometimes, it is not possible to avoid human interaction completely due to the complexity of an experimental dataset, or the relationship discovery and integration process.

9.5.4 transparency and autonomy

IDMBD shields users from the underlying structure of sources. The end user of the integration system does not need to know the underlying structure of sources when accessing or querying the heterogeneous data sources. This is achieved by using the mediator/wrapper technology. The mediator uses wrappers that encapsulate the underlying structure of data sources, so that wrappers' access to data sources is transparent to the mediators. This preserves a data source's autonomy and gives a biologist access to these sources, and enables him/her to retrieve the most up-to-date biological data.

9.5.5 Flexibility

IDMBD makes it easy for a biologist to link and analyse experimental datasets. It allows easy integration of a dataset, using different biological relationships with public data sources via different relationships, and linkage approaches, thus, providing the ability to use different relationships, linkages and threshold values. It is also flexible in terms of its ability to link datasets with other datasets, link datasets with public data sources, or link public data sources with other public data sources.

9.5.6 Extendibility

The IDMBD system's architecture allows for extendibility by the addition of new data sources to the integration system. With little effort, a new data source can be plugged into the system, by:

vi) Data registry: this registers a new data source's information, i.e., name, location, wrapper, access information.

vii) Schema manipulation: this involves creating a source schema definition, importing its metadata and then mapping the local schema to the IDMBD's global schema.

viii) SLM and RKB: relationships between the new source and existing sources are discovered to build the SLM (see section 5.3) and the relationship tables are generated and added to RKB.

ix) Wrapper: generates a wrapper for the new source.

9.5.7 Heterogeneity

IDMBD overcomes heterogeneity by using a relationship to integrate data. This is achieved by SLM, which allows the species-specific data sources to be linked without problems due to name clashes and ambiguities. Moreover, experimental dataset concepts are mapped to a Domain Ontology, which also helps to resolve heterogeneity. The wrapper handles all other heterogeneity conflicts. For example, Arabic numbers are used to represent chromosomes in the mouse data source (MGI) and Roman numerals are used in the *C. elegans* data source (Wormbase). A mapping is used to resolve this type of heterogeneity in the wrapper.

9.5.8 Functionality

IDMBD supports different types of queries, such as single search, multiple search, and links datasets for a specific species with datasets from other species as well as linking these datasets with public data sources. The key to this in IDMBD is the alternative relationships provided to discover new knowledge across species. Using these different relationship types for linkage allows a biologist to obtain different result sets. This allows different aspects to be investigated, so providing the biologist with useful information about the genome. These different result sets provide gene enrichment as illustrated in section 8.5. The quality of the datasets obtained by the system and the final validation of a biologist hypothesis has shown the value of this approach to biologists.

9.5.9 Original Goals Revisited

Based on the problem specification we set out in chapter 1 and 3, we have achieved the original objectives we sought to address. In particular, we have achieved the following:

- Developing the IDMBD system that allows a bioinformatician to extract an experimental dataset's metadata, detect suitable candidate keys for the linkage (Objective 1) in order to link the experimental dataset with public bioinformatics data sources, and transform the extracted metadata and datasets into a form that can be used for linkage with other sources (Objective 2) These tasks have been successfully undertaken and demonstrated in Chapter 5, 6, and 7.

- Using the biological relationships to provide flexible and loosely coupled linkages across heterogeneous data sources (Objective 3) was achieved through the SLM approach as discussed in section 9.4.1.

- Building a knowledge base of discovered relationships between sources (Objective 4). This has been done successfully by building RKB (see section 9.4.1).

- The IDMBD system provides users with uniform access to the bioinformatics sources and shields users from the underlying structure of sources (Objective 5) as discussed in section 9.5.2 and 9.5.4.

9.6 Summary

In this chapter, we evaluated the approach with respect to its primary aims and objectives. We demonstrated the main advantages in terms of time and genericity, resolving heterogeneity and minimizing human interactions.

We setup the Relationship Knowledge Base (RKB) to store several relationships across species. This knowledge base stores the biological relationship type, and the relationship closeness between biological objects across species. Once this knowledge base is created, the system uses it to link and compare datasets across species. This set up overhead for RKB occurs once when comparing entire genomes across species. The subsequent experiments are then analyzed, linked and compared easily and quickly since the system uses the existing stored relationships in RKB in subsequent investigation; thus, there is no need to perform these comparisons for each a new experiment. Our system saves user preparation time by the automation of manual tasks occurring in several processes. These manual tasks are prone to error particularly if the researcher is interrupted by the phone or by colleagues. Therefore, the mistakes and errors will be high. For example, when a researcher resumes an analysis after a break, he/she may forget which datasets they were using or mistakenly use different datasets, while this is not a problem with our automated process. In our system, a user only supplies experimental datasets and sets up his/her parameters. Thereafter, the system does the rest of the process: processing experimental datasets, extracting metadata, converting to a suitable format, linking to public sources, retrieving data, using relationship knowledge to link to other species and comparing across species and mapping the result to a unified format.

In the next chapter, we discuss the overall conclusion of this work and consider some ways in which the framework can be extended.

Chapter 10

Conclusions and future work

10.1 Synopsis

We draw conclusions about the research and identify future work that can be undertaken to take this research forward. A summary of the work reported is given, with a discussion of the extent to which contributions have been made. In addition, the currently perceived strengths and limitations of these contributions are outlined, followed by suggestions about possible areas of future research directions.

10.2 Thesis summary

In this thesis, we have presented a novel approach to interoperability based on the use of biological relationships that use relationship-based integration to integrate bioinformatics data sources. This involves using different relationship types with different relationship closeness values to link gene expression datasets with other information available in public bioinformatics data sources. These relationships provide flexible linkage enabling biologists to discover linked data across the biological universe. Relationship closeness is a variable used to measure the closeness of the biological entities in a relationship and is a characteristic of the relationship. The novelty of this approach is that it allows a user to link a gene expression dataset with heterogeneous data sources dynamically and flexibly to facilitate comparative genomics. Our research has demonstrated that using different relationships gives the user a better understanding of the genomic functions of genes as it adds biologically rich information derived from different bioinformatics data sources to the gene lists obtained from experiments.

Our survey of biological and bioinformatics literature found the more important relationships between biological objects are homolog, ortholog, paralog, biological process, cellular component and molecular function, so we developed the system to link information across species based on these relationship types.

In an experiment, we applied our system to two different sets of data related to growth and aging in two different species. First, the system extracted metadata from these experimental datasets, created a schema and then converted it to a suitable format (relational). Then it nominated a candidate key to be used for linking these datasets with public data available to the user. The linkage key was then mapped to a domain ontology to extract related concepts and relationships. Finally, the system linked the experimental dataset with public sources using the soft linkage approach. For each experiment, we used different types of linkage (relationship type). Then we ran our system with the same datasets several times with different relationships each time. This gave different result sets, which reflected how the biological objects were connected with each other in different ways. These different results allowed a biologist to analyse the datasets in different ways and gave insight into the nature of biological objects. These processes enabled the formulation of novel hypotheses by the biologist leading to the informed design of new cycles of laboratory research. Moreover, a measure of relationship closeness should give a biologist a new tool in their repertoire for analysis. Thus, these experiments have shown how we can use SLM to link a dataset with public data sources in different ways using the relationships to provide data integration within the framework of a data analysis process, and that:

- The data generated suggests that the functional cross-species interpretation using an orthology model must be informed by the specific inter-species relationship between the orthology and function.

- It is clear from the data that by exploiting a variable threshold level we can define a cross species mapping, which is extremely conservative, either by identifying an orthology group that has a maximal probability of sharing function or by selecting a lower threshold whereupon we obtain the maximum return of related genes and still minimise the noise generated by random matches.

10.3 Thesis contributions

The following is a summary of the achievements of the research:

- Introduction of a new approach to extracting an experimental dataset's metadata and identifying the most appropriate candidate key for linkage with other related data. The thesis describes an approach to automatic text extraction, in particular the identification of biologically-relevant fields in a flat file. The extraction of this information allows a user to link and integrate the data parsed from a flat file automatically with public resources such as Wormbase, Swiss-Prot, Gene Ontology and others.

- Introduction of a novel approach to the integration of bioinformatics data sources, which allows a biologist to investigate easily alternative linkages. This approach allows a biologist to integrate and link experimental datasets that can be used for the rapid functional annotation of genomes with available public specific-genome repositories. Our approach was a relationship-based query and integration process rather than a key-based integration and query approach. Thus, the integration is based on the relationships between properties of concepts not field-values. In addition, one of the features of our approach is that the user can customize how the data sources are linked by building his/her own SLM. This allows the use of different relationships with different relationship-closeness values to link gene expression datasets with other information available in public bioinformatics data sources.

- An improvement in comparative approaches to annotating genes, by identifying possible relationships between objects across species, and predicting protein-function from sequence homology, orthology and GO-terms. By integrating functional and sequence data across species, we can annotate the genome of a species using functional data from another genome. Comparative genomics provides evidence for close evolutionary relationships between gene families. This is implemented in our system by building a knowledge base of the discovered relationships between biological objects, which is used to compare and link the experimental datasets with public sources. This has been verified through the creation of the RKB (see Chapter 7) to capture the semantic relationships (homology, related molecular function, related biological process and related cellular component) between genomic data across species in a way that allows integration across species.

- Determining the optimal threshold for cross-species orthology relationships. This is demonstrated for Mouse and C.elegans (Section 8.5).

10.4 Strengths and Limitations of SLM

The key aspects of the SLM approach are:

- SLM integrates data from remote sources without bringing the data physically into a central database when the researcher needs it. Thus, it uses the current version of the data in the public sources.

- The biologist can change the linkage type according to the research agenda. Depending on the research questions being asked, the biologist can choose appropriate linkage between the different concepts and objects. They may easily investigate different linkages to determine what they reveal. This is due to the system's flexibility and support for different types of linkages.

- The SLM provides linkage of genetic databases to other databases, including non-bioinformatics databases, containing information about concepts such as drugs, biochemistry, clinical information and patients. For example, a clinical database may not have a one-to-one mapping with a genetic database, but there is a clear relationship, which can be presented in SLM. This means the system is extendible.

- The SLM stores relationships between sources in a Relationship Knowledge Base (RKB) and exploits them to combine annotation knowledge from different data sources. The RKB can be exchanged and reused.

- The SLM allows a user to browse the discovered relationships between data sources, and the objects involved in a specific relationship.

- A user can customise the linkage between an experimental dataset and one or more public sources by customising the SLM.

- The IDMBD prototype system was implemented as an illustration of concept prototype with only two species currently supported. It can be extended by registering new sources and building the SLM.

- The prototype system accepts only delimited flat files where the first line contains column names or headers.

10.5 Future Work

The research in this thesis has generated many interesting and promising ideas. Some of these are worth exploring further. In this section, we describe several directions for future research.

- Consideration of other relationship types like pathways and synteny. Pathways would give greater insight into how the protein works and would assist in the discovery of new drugs. Synteny would help predict the location of new genes by comparing uncharacterized region with a characterized region in another genome.

- Insight into the flexibility obtained from this study should be used to extend the system to enable integration of non-bioinformatics data sources with bioinformatics sources, for instance, medical data sources, via different semantic relationship types. For example, this approach could be used in medical genetics to find relationships between a specific disease and genetic structure. This could help scientists to design new drugs for a disease.

- Future research can look at comparing the outcome of using different techniques to calculate GO-based similarity.

- Future work can look at the interpretation of the wealth of the relationships in RKB to predict gene-product (protein) function.

- In this work, we did not consider schema integration; thus, the global schema is specified by the integrator. It should be possible to semi-automate the process of constructing a global unified schema that characterizes the underlying data sources.

- In metadata extraction, we considered only extraction from flat tab-delimited/comma-delimited types of file. A more general solution of this problem would be useful because it would allow structured databases to be created automatically from various experimental datasets.

10.6 Conclusion

In conclusion, this thesis covers an approach to the integration of diverse bioinformatics data sources using a flexible linkage. It is a novel approach that provides a flexible and soft linkage between data sources. Soft Links are modelled via concepts that are interrelated, using a rich set of possible relationship types. Such a flexible relationship allows biologists to mine effectively the exponentially increasing amount of comparative genomic information. This can be used as a basis to enable cross species functional annotation of data generated by array experiments to inform better the selection of targets for more detailed analysis based on cross species functional information. Furthermore, once the SLM are established, secondary analysis on genomic elements such as the transcription control elements (transcription factor binding sites) can be analysed to provide novel insights into the evolutionary conservation of gene expression. By integrating functional and sequence data across species, we are able to annotate the genome of a species using functional data from the other species, as comparative genomics provides evidence of close evolutionary relationships between gene families.

Finally, the key concept embodied in IDMBD that differentiates it from other systems is its use of semantic relationships between biological objects to link data across heterogeneous data sources in a flexible manner. To the best of our knowledge, no existing system integrates gene expression datasets with publicly available bioinformatics data sources to facilitate comparative genomics in such a flexible way. This enables a biologist to obtain more understanding of genes and their functionality.

System comparison

This appendix provides a comparison of the bioinformatics integration system according to different dimensions.

System	Integration Approach	Data model	Level of transparency	Integration Degree	materialization	Data types	Query operators
SRS	Data warehousing	Icarus N/A	Sources specified by user	Loose	completely *materialized*	Strings, NA, AA seq.	Boolean pred., reg. exp, homology search
K2/Bio-Kleısli	Federation	complex value model; Object-Oriented	Sources specified by user	Loose	views	String	SQL
BACIIS	Mediation	XML	Sources selected by system	Loose	views	Strings	N/A
KIND	Mediation	XML(data) F-Logic (CM)	Sources specified by user	Loose	views	String	Boolean pred., homology search
BioDataServer	Mediation	Relational	Sources specified by user	Tight	completely materialized	String, integer,partially NA, AA seq.	SQL
Discovery-Link	Federation	Relational	Sources selected by system	Tight	views	Strings	Read only SQL
GUS	Data warehousing	Relational	N/A	Loose	Views	Strings, NA, AA seq.	Boolean pred., homology search
Bio-Navigator	Data warehousing	Unstructured text model	Sources specified by user	Loose	Views	Strings, NA, AA seq.	Boolean pred., homology search
Entrez	Data warehousing/web-based navigational	ANS.1	Sources specified by user	Loose	views	Strings, NA, AA seq.	Boolean pred., homology search
TAMBIS	Multi DB queries	Descriptive logic. GRAIL	Sources hard-wired by system	Loose	views	Strings	Case based qrs. ontologies
ISYS	Federation	Object-Oriented	Partial	Tight	views	Strings	Read only SQL
LIMBO	Data warehousing	Relational	Sources hard-wired by system	Tight	views	Strings	Boolean pred., homology search
GenMapper	Data warehousing	Relational(GAM)	Source specified by user	Loose	views	String	Boolean pred.,reg. exp
SEMEDA	Mediation	Relational	Sources specified by user	Loose	views	String	Boolean pred
P/FMD	Mediation	functional data model, object-oriented	Sources specified by user	Loose	views	String	Boolean pred
TINet	Multi DB queries	Object-Relational (OPM model)	Sources specified by user	Loose	views	String, NA,AA seq.	Boolean pred.,reg. exp, homology search
ALADIN	Data warehousing	Relational	Low	Loose	materialized	String	All SQL operators
BIS	mediation	Relational	Low	Loose	view	string	All SQL operators

APPENDIX A: SYSTEM COMPARISON

EMBL Harvester	Data warehousing	OO	Low	Loose	views	String	Boolean pred.,reg. exp, homology search
EnsEMBL	Data warehousing	OO	Sources specified by user	Loose	views	Strings, NA, AA seq.	Boolean pred.,reg. exp, homology search
GenoMax	Data warehousing	Relational	Sources specified by user	Tight	view	String	Boolean pred.,reg. exp, homology search
OPM	multiDB query	Relational; OO	Sources specified by user	Loose	views	Strings, NA, AA seq.	
INDUS	Federation	Relational	Fully transparent to the end-user	Loose	views	A hierarchical type system based on user and data source ontologies	Relational, statistical
BioMediator	Mediation	RDF	Sources specified by user	Loose	view	Strings and URLs	Regular expressions, Conjunctive queries
COLUMBA	Data warehousing	Relational	Sources specified by user	Loose	materialized	Strings	Boolean pred.,reg. exp,
Pegasus	Federation	Iris Object-Oriented Model	Sources specified by system	Loose	views	Strings	Boolean pred.
TSIMMIS	Mediation	OEM Object Exchange Model (in OO)	Partial transparent	Loose	views	Strings	Boolean pred.
SIMS	Mediation	Description logic (LOOM)	Sources specified by system	Loose	views	Strings	Boolean pred.
Garlic	Mediation	OO-like	Sources specified by user	Loose	views	String, multimedia	All SQL operators
DAVID	Data warehousing	Relational	Sources specified by user	Loose	view	String	Boolean pred.,reg. exp, homology search

APPENDIX A: SYSTEM COMPARISON

System	User model	Data Source interface	Global Schem a	Number of sources	Resolving heterogeneity	Domain	Ontology	Query planning
SRS	No critical expertise. Simple to use visual interface	Declarative language(Icarus)	No	43	No	Biological Data	No	N/A
K2/Bio-Kleisli	Require knowledge of SQL	Wrapping mechanism= CPL	Yes	60	No	Biological Data	No	Query optimizer, Cost-based
BACIIS	Interactive query formulation	Wrappers, java	Yes	7	Yes	Biological and chemical	BAO	Adaptive, SQL based, serially ordered sub-queries. (GraphPlan)
KIND	Expertise in query language	wrappers	Yes	N/A	Yes	Neuroscience	Yes	Query decomposition using Domain Ontology
BioDataServer	No critical expertise	wrappers	Yes	3	Yes	Genome, pathways	No	mechanisms for query decomposition and data source localization
Discovery-Link	Expertise in query language	Wrappers using C++	Yes	arbitrary	Yes	Life sciences	No	Query optimizer
GUS	Expertise in query language	wrappers	Yes	5	Yes	Genomics data	Yes	N/A
Bio-Navigator	No critical expertise	wrappers		arbitrary	No	Sequence and Structure analysis	No	Predefined execution path
Entrez	No critical expertise	wrappers	Yes	> 20	No	Molecular biology	No	N/A
TAMBIS	Interactive query formulation	Wrapping mechanism= CPL	No	>6	Yes	Biology, focus on protein and nucleic acids	TaO	*search algorithm +mapping collection. * ordered query components list * map query components to functions
ISYS	Interactive discovery	wrappers	No	arbitrary	Yes	Genome	No	Complex query
LIMBO	Interactive query	wrappers	No	3 (arbitrary)	No	Genetics	No	No
GeneMapper	Simple to use visual interface	wrappers	No	7	No	Genetics	GO	N/A
SEMEDA	Simple to use visual interface	wrappers	Yes	Arbitrary	Yes	Biology	Yes	N/A
P/FMD	Simple to use visual interface	Wrapper	Yes	Arbitrary	No	Protein	No	N/A
TINet	Simple to use visual interface	Wrapper using SDK	No	6	No	genomic sequences	No	N/A
ALADIN	No critical expertise	Wrappers- a relational representation of the source database	No	15	Find objects links between different sources: explicit contained links (sources that reference other ones) and implicit contained links (by looking for similar values)	Life sciences	No	Utilize the database system

					and duplicate objects between sources			
BIS	No critical expertise	wrappers	Yes	3	No	Biological data	No	optimizer
EMBL Harvester	Simple to use visual interface	wrappers	No	10	No	human proteins	No	N/A
EnsEMBL	Simple to use visual interface	wrappers	Yes	3	No	Eukaryotic genomic sequence	Yes (GO)	N/A
GenoMax	Need expertise	wrappers	Yes	N/A	Yes	biological sequence data, gene expression data, 3D protein structures, and protein-protein interaction data		Yes with a proprietary scripting language
OPM	No critical expertise; expert user use OPM*QL	C/C++ API	Yes	3	Yes	biomedical		
INDUS	Yes (user ontology and user-specified mappings)	wrappers	No	As many as one likes	Yes-using mappings from user ontology to data source ontologies	Yes	Yes	Yes
BioMediator	Simple	wrappers	Yes	20	Syntactic and semantic	Genetics, molecular biology, anatomy and neuro-informatics	Yes. For data, mediated schema and for source annotation	Only for path generation
COLUMBA	Simple	wrappers	Yes	7	Yes to some extent	Gene annotation, Protein Structure Annotation	GO	N/a
Pegasus	User should be able to use standard terminology to compose HOSQL query	wrappers	Yes	arbitrary	Yes - resolve conflicts in naming. structures, and data domain	arbitrary	No	Cost-based optimization
TSIMMIS	simple	Wrappers with high level description language	No	Arbitrary	No	Independent	No	Views templates, combines several views to answer complex query
SIMS	User should be able to use standard terminology to compose LOOM query	wrappers	Yes	Arbitrary	Yes	Independent	Yes	AI planner UCPOP 2.0, partial-order planning. The query processing mechanism based on planning can determine a very complex relationship between the collection of information requested by the user and the data available from the various

								sources. planning by rewriting (PbR) approach
Garlic	Simple	DB wrappers	Yes	arbitrary	Yes	Large-scale multimedia information system	No	Parsing, semantic checking, query rewrite and query optimization
DAVID	Simple	Wrappers =java & perl	Yes	9	No	Functional genomic annotation	No	Parsing, semantic checking, query rewrite and query optimization

System	Query caching	Query adaptive	System platform	Domain schema	User interface	Query language	API	Output format
SRS	Yes	No	Web	Icarus	Web-based, HTML	N/A	C - API	HTML, ASCII
K2/Bio-Kleisli	Yes	Yes	Java	ODL	Text-based, RMI	OQL	RMI, JDBC	various
BACIIS	User queries, query plans, selected results	Multiple query cycles with adaptive planning	C++/Java with COBRA interface on UNIX	Ontology using CLASSIC, data model and schema using XML	Web-based, HTML,XML,JSP	N/A	N/A	HTML
KIND	Yes	Query optimization using domain knowledge	Java	XML DTD	Web-based	F-Logic	N/A	HTML
BioDataServer	N/A	N/A	Java	JavaCC grammar	Web-based, java applet	SQL	ODBC,JDBC	Various
Discovery-Link	Yes		Independent	Relational		SQL	JDBC	Various
GUS	N/A	N/A	Unix	Relational	JSP,PHP	SQL	PERL API	HTML
Bio-Navigator	Yes	Yes			Web-based	N/A	N/A	HTML
Entrez	Yes				Web-based	-	N/A	HTML, ASCII, XML, ASN.1
TAMBIS	Yes	No	Java	Ontology in GRAIL, BioCon KB	Java applet	CPL	N/A	TEXT
ISYS			Java	Relational		SQL	JDBC, ODBC,CORBA	Various
LIMBO	No	No	Java	Relational	Web-based	SQL	N/A	HTML
GeneMapper	No	No		Relational	Web-based	N/A	N/A	HTML
SEMEDA	No	No	Web	Relational	Web-based	SQL	N/A	HTML
P/FMD	N/A	N/A	Java		Prolog and the Daplex interfaces, web-based, Java-based	DAPLEX query	N/A	Text, HTML
TINet	N/A	N/A	Java		Web-based, HTML,	SQL-like	C++API,CORBA	N/A

					command-line			
ALADIN	Utilize the database system	N/A	All RDBMS	All schemas remain independent - there is no specific domain schema	Web-based, HTML	SQL	N/A	HTML
BIS					Web-based	N/A	N/A	N/A
EMBL Harvester	N/A	N/A	Web		Web-based	None	None	HTML, EXCEL
EnsEMBL	N/A	N/A	Web		Web-based	N/A	N/A	Various
GenoMax	Yes with a proprietary scripting language	with the creative scripting	Sun 15K with Solaris 9 using Oracle, Windows		Java-based Graphical User Interface	N/A	N/A	HTML or (tab-text plain text for .XLS)
OPM	Yes	N/A	Independent		Web-based	SQL	CORBA	Various
INDUS	No	No	any platform supporting JDK 1.4 or above.	Yes through domain ontologies	Stand alone Java application	Ontology-based relational, extended with statistical operators	N/A	flexible
BioMediator	At the metawrapper level	Using Tukwila	java		Various by application developer	PQL(v1), attribute/value paris(v2)	POL on sockets (v1) and Java methods(v2)	XML, RDF,HTML
COLUMBA	N/A	N/a	Web	Relational	Web-based	SQL	N/A	XML, text, HTML
Pegasus	Yes	Query optimizer	Independent	Object Oriented	Various	HOSQL	N/A	Text
TSIMMIS	No	OEM DB(LORE) cache OEM objects. LOREL was used to query LORE	Independent	No	MOBIE WWW page, query menu, hypertext answer	OEM-QL (LOREL)	N/A	HTML
SIMS	Yes	Selectively materialising data by analyzing query distribution, source structure and maintenance cost	Independent	Objects and objectives LOOM	Web-based, form-based	LOOM	N/A	CLIM
Garlic	No	Query-optimizer, cost-based	C++ API	OO-Like	PESTO; interface, friendly, browsing, navigation	SQL	C++API	HTML
DAVID	No		Java	relational	Web-based interface	N/A	N/A	HTML,TEXT

XML documents and Schema

This appendix contains the schema of the data sources and an example of data sources metadata.

Element	**description**
ID	identifier of the data source
Name	name of the data source
Description	description of data source
Owner	data source owner, if supported
URL	A specialized form of URL is used by JDBC to identify databases.
System	the system in which data source is running
Database Type	type of the data source management system.
Direct Access	flag to indicate whether there id direct access or not to the data source
Host	IP address of the data source.
Port	Port number to be used to connect.
User Name	user name to access the data source
Password	password to access the data source
JDBC_DRIVER	A Java class that implements the JDBC driver interface and is loaded into the JDBC driver manager.

Table B.1: Description of xml schema elements

Metadata Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="ID" type="xs:string" minOccurs="1" maxOccurs="1"/>
 <xs:element name="Name" type="xs:string" minOccurs="0" maxOccurs="1"/>
 <xs:element name="Description" type="xs:string" minOccurs="0"
maxOccurs="1"/>
  <xs:element name="Owner" type="xs:string" minOccurs="1" maxOccurs="1"/>
  <xs:element name="URL" type="xs:integer" minOccurs="0" maxOccurs="1"/>
  <xs:element name="System" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="DataBase" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="Direct_Access" type="xs:boolean" minOccurs="1"
                    maxOccurs="1"/>
  <xs:element name="Host" type="xs:string"  minOccurs="1" maxOccurs="1"/>
  <xs:element name="Port" type="xs:integer" minOccurs="1" maxOccurs="1"/>
  <xs:element name="User Name" type="xs:string" minOccurs="0" maxOccurs="1"
                    />
  <xs:element name="Password" type="xs:string" minOccurs="0" maxOccurs="1"/>
  <xs:element name="JDBC_DRIVER" type="xs:string" minOccurs="0"
                    maxOccurs="1"/>
 </xs:schema>
```

Figure B.1: XML schema of metadata of data sources

Data sources description

```
<?xml version="1.0" standalone="yes"?>
<Databases>
<Database>
        <ID>DB1</ID>
        <NAME>Wormbase</NAME>
        <DESCRIPTION>WormBase is the repository of mapping, sequencing and
          phenotypic information for C. elegans (and some other
        nematodes)</DESCRIPTION>
        <OWNER>Sanger Institute</OWNER>
        <URL>www.wormbase.org</URL>
        <SYSTEM>DataBase Managemeny System></SYSTEM>
        <DATABASE_TYPE>AceDB</DATABASE_TYPE>
        <DIRECT_ACCESS>true</DIRECT_ACCESS>
        <HOST>aceserver.cshl.org</HOST>
        <PORT>2005</PORT>
        <USERNAME>anonymous</USERNAME>
        <PASSWORD>****</PASSWORD>
        <JDBC_DRIVER_NAME></JDBC_DRIVER_NAME>
        </Database>
<Database>
        <ID>DB2</ID>
        <NAME>Mouse Genome Informatics (MGI)</NAME>
        <DESCRIPTION>Mouse Genome Informatics (MGI) provides integrated
        access to data on the genetics, genomics, and biology of the laboratory
        mouse.</DESCRIPTION>
        <OWNER>The Jackson Laboratory</OWNER>
        <URL>http://www.informatics.jax.org</URL>
        <SYSTEM>DataBase Managemeny System></SYSTEM>
        <DATABASE_TYPE>Sybase DB</DATABASE_TYPE>
        <DIRECT_ACCESS>true</DIRECT_ACCESS>
        <HOST>gondor.informatics.jax.org</HOST>
        <PORT>4025</PORT>
        <USERNAME>badr</USERNAME>
        <PASSWORD>****</PASSWORD>
        <JDBC_DRIVER_NAME>com.sybase.jdbc2.jdbc.SybDriver</JDBC_DRIVE
        R_NAME>
</Database>
</Databases>
```

Figure B.2: Metadata description of data sources

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
  XML schema for Soft Link Model metadata.
 </xsd:documentation>
</xsd:annotation>
<xs:element name="SLM-knowledge-base">
<xs:attribute name="no" type="integer" use="required"/>
<xs:element name="database" minOccurs=0 maxOccurs="unbounded">
<xs:complexType>
<xs:element name="concept" minOccurs=0 maxOccurs="unbounded">
 <xs:complexType>
<xs:element name="relations" minOccurs=0 maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
 <xs:element name="SLM" minOccurs=1 maxOccurs="unbounded">
 <xs:attribute name="DBName" type="RC" use="required"/>
 <xs:attribute name="concept" type="String" use="required"/>
<xs:attribute name="RelationType" type=" relationships " use="required"/>
<xs:attribute name="File" type="String" use="required"/>
<xs:attribute name="FileType" type=" String " use="required"/>
</xs:sequence>
</xs:complexType>
</xs:complexType>
</xs:complexType>
<- ->
<xsd: simpleType name="relationships">
<xsd:restriction base="xs:string">
<xsd:enumeration value="homolog"/>
<xsd:enumeration value="ortholog"/>
<xsd:enumeration value="MolecularFunction"/>
<xsd:enumeration value="BiologicalProcess"/>
<xsd:enumeration value="CellularComponent"/>
</xsd:restriction">
</xsd: simpleType>

<xsd: simpleType name="filetype">
<xsd:restriction base="xs:string">
<xsd:enumeration value="mySQL"/>
<xsd:enumeration value="text"/>
<xsd:enumeration value="OO"/>
</xsd:restriction">
</xsd: simpleType>
</xs:schema>
```

Figure B.3: XML schema for SLM metadata

Technologies

The technologies used in the implementation of IDMBD system are summarised in Table C.1 with reason for use.

Technology	reference	reasons
JavaBeans	http://java.sun.com/products/javabeans/	JavaBeans are reusable software programs that can be developed and easily assembled to create sophisticated applications.
JavaServer Pages	http://java.sun.com/products/jsp/	JSP is a server-side technology that is an extension of the Servlet technology. It facilitates the creation of web applications that have both static and dynamic components. It supports the use of JavaBeans components with standard JSP language elements.
Java Servlets	http://java.sun.com/products/servlet/	Servlets are the preferred Java platform technology for extending and enhancing the functionality of a Web server. They provide a component-based, platform-independent method for building Web-based applications and have access to the entire family of Java APIs, including the JDBC API to access enterprise

		databases.
BioJava	http://www.biojava.org, Article: BioJava -- Java Technology Powers Toolkit for Deciphering Genomic Codes *By Steven Meloan, June 2004,* BioJava: open source components for bioinformatics; Matthew Pocock	BioJava is an open source Java Library for bioinformatics designed for providing a Java framework for processing biological data.
BioPerl	www.bioperl.org	The Bioperl project is an international open-source collaboration between biologists, bioinformaticians and computer scientists whose aim is to build bioinformatics solutions in Perl and to provide a comprehensive library of Perl modules for managing, handling and manipulating life science data.
AcePerl	http://stein.cshl.org/AcePerl/	AcePerl, written by Lincoln Stein, is an excellent object-oriented Perl interface module providing virtually transparent access to local or remote ACeDB databases, performing queries, fetching ACE objects, and updating databases
Tomcat Server	(http://tomcat.apache.org/)	The Tomcat server is an open source, free to use, Java based Web Application container created to run Servlets and JavaServer Pages (JSP) in Web

		applications.
Apache	http://www.apache.org/	The Apache HTTPD server is a powerful, flexible, HTTP/1.1 compliant web server that implements the latest protocols, including HTTP/1.1
Java 2 SDK	http://java.sun.com/j2se/1.4.2/docs/index.html	The essential Java 2 SDK provides tools, runtimes, and APIs for developers writing, deploying, and running applets and applications in Java programming language.
Mod_Jk	http://tomcat.apache.org/connectors-doc/	Mod_Jk is the Tomcat-Apache plug-in that handles communication between Tomcat and Apache.

Table C.1: technologies used in the implementation of IDMBD

Classes

This appendix provides samples of Java classes used in the implementation of the IDMBD system.

```
Public Class SoftLinkAdaptor() {

 Public Vector getRelations(String SLM)

 Public Vector GetMatchEntriesInDataSource(Sring id, vector matchentry, string db,
string concept)

 Public Vector GetMatchEntriesInDataSource(Vector id, vector matchentry, string db,
string concept)

 Public Vector GetMatchEntriesInRelationTable(Sring id, String dataSource, String
relationfilename)

Public Vector getRelation(String db, String concept)

Public Vector getOther(Vector result, String db, String concept, String condition)

Public Vector getOther(Vector result, String db, String concept);

}
```

Figure D.1: Main SoftLink Interface Class with Primitives for SLM API

```
Public Class QueryHandler (

        Public ExtractMetadata(String filename)

        Public ExtractMetadata(String filename, String delima)

        Public double ComputeScore(Vector dataset)

        Public Boolean isKey(String tag)

        Public Boolean isAmbiguous(Vector dataset)

        Public Boolean isNull(Vector dataset)

        Public Boolean isSingleValue(Vector dataset)

        Public Boolean isUnique(Vector dataset)

        Public String DataType(Vector dataset)

        Public int elementLength(Vector dataset)

}
```

Figure D.2: Query Handler Class with Primitives for SLM API

```
public Class RelationshipWrapper {

        public Vector getRelationshipId(String key, String tableName) ;

        public Vector getRelationshipId(String key, String tableName, double e_value,

                        int score, double rc) ;

        public Vector getRelationshipId(String key, String tableName, String condition);

        public Vector getRelationship(String key, relationsInfo rl) ;

}
```

Figure D.3: RelationshipWrapper Class with Primitives for SLM API

```
    Public Class  GenerateSoftLinkTable {

        private Map loadAlgorithms(String algXMLfile)

        private void saveRelationshipTable(java.util.List entries)

        private void CmdExec(String cmdline)

        private Map getAlgorithm()

        private Algorithm getAlg(Vector v, String name)

        private void run_algorithm(String s1, String s2, String alg, String output)

        private String formatPath(String cmd)

    }
```

Figure D.4: GenerateSoftLinkTable Class

```
  Public Class BlastParser {

      Class BlastLikeSAXParser

      Class SeqSimilarityAdapter

      Public Vector getblastParser(String filename)

      List getBlastParser(String filename)

      }
```

Figure D.5: BlastParser Class with Primitives for SLM API

```
public Class Gene {

 public class MapPosition {

  public String ChromosomeNumber;

  public String centimorganPosition;

  public String cytogeneticOffset;

 }

 public Class DBlinks {

  public String UniGene;

  public String LocusLink;

 }
```

Figure D.6: Gene Class

```
public Class relationsInfo {

 public String rootDbName;

 public String rootConceptName;

 public String DbName;

 public String Concept;

 public String RelationType;

 public String RelationFile;

 public String FileFormat;

}
```

Figure D.7: relationsInfo Class

```
public Class Algorithm {

 String name;

 String location;

 String syntax;

private static Algorithm getParameters(String name)

private static Map getAlgorithm()

}
```

Figure D.8: Algorithm Class with Primitives for SLM API

```
public Class UniGeneWrapper {

        public void connect();

        public void close();

        public int getNumberOfRecords();

        public String getUniGene(String AccId);

}
```

Figure D.9: UniGeneWrapper Class with Primitives for SLM API

```
Public Class WrapperManager() {
    public void getWrapperName()

    public Vector getWrappers()

    public void links(keys,db,concept)

    public void SoftLinkCallBack():

}
```

Figure D.10: Wrapper Manager Class

```
Public Class Wrapper() {

// returns gene entries for a specific gene using a specified identifier

fetchRecord(String id)

// returns gene entries for multiple genes using a specified identifier.

fetchRecord(Vector ids)

// returns gene entries for multiple genes using a specified identifier.

fetchRecords(Vector ids)

//returns gene entries for multiple genes using a specified search field.

fetchRecords(Vector ids, String SearchKey)

}
```

Figure D.11: Wrapper Class

```
public Class GOWrapper {

        //it connects to the data source

        public void connect();

        // it closes all connections to the database and releases resources reserved for
        the connection.

        public void close();

        // returns number of records.

        public int getNumberOfRecords();

        //fetches a GO entry for a specific accession number.

        Public String geGO(String AccId);

}
```

Figure D.12: GOWrapper Class

```
public Class SLMParser{

//parses a SLM and loads relationships in a hash table.

parse(Stringfilename)

//gets all relationships from a hash table.

getAllRelationship(Hashtable slm)

//gets all relationships of a concept from a data source

getRelation(Hashtable slm, String db, String concept)

}
```

Figure D.13: SLMParser Class

Biologist's Evaluation

This appendix includes the evaluation letter from Dr. Peter Kille (*Bioscience School, Cardiff University*). He used the system and was impressed by the findings.

Cardiff School of Biosciences
Head of School Professor J L Harwood PhD DSc

Ysgol y Biowyddorau, Caerdydd
Pennaeth yr Ysgol Yr Athro J L Harwood PhD DSc

Cardiff University
Biomedical Sciences Building
Museum Avenue
Cardiff CF10 3US
Wales UK

Tel +44(0)29 20 874108
Fax +44(0)29 20 874117
Email Harwood@cardiff.ac.uk

CARDIFF
UNIVERSITY
PRIFYSGOL
CAERDYDD

Mr Al-Daihani,
School of Computer Science,
Cardiff University,

30/07/2008

Evaluation of Soft Link Model Performance

Dear Mr Al-Daihani,

I was extremely intrigued to receive the cross species comparison of genes associated with aging generated by the Soft Link Model (SLM). In my opinion the data provides some biologically relevant insights both generally, in the context of the relationship between functional conservation and homology, together with more specific insights realised through identification of evolutionarily conserved age related genes.

The issue of homology threshold and its relationship to gene function is critical when performing inter-species comparisons. However, the majority of studies use generic values based on solely of the statistical probability of a sequence homology occurring by random (Blast E-value) without any knowledge of the relationship between this statistical value and functional conservation which will be specific to genetic divergence between the two species being studied. The results generated by SLM which compares the proportion of genes with conserved functional ontological definitions, for biological process, molecular function and cell component, under various degrees of homology shows an extremely interesting relationship. Intriguingly, it revealed a biphasic function justifying the accepted homology threshold or E-10 as being appropriate to yield an inclusive set of functionally related genes whilst a probability score >E-40, although yielding substantially fewer genes, provides a much higher confidence in functional conservation. This analysis is extremely useful when mining cross disciplinary data sets between these two species and demonstrates the power of generating similar analysis for other cross-species comparisons a process which would be substantially stream-lined should the SLM interface be expanded to include primary data sources for additional species.

The two studies identifying age related transcript changes illustrates a generic challenge facing many global analysis approaches, that being the shear number of responsive genes identified. One approach allowing targeting of further experimental work is to identify responses which are evolutionarily conserved. The SLM analysis of these data sets provides an elegant illustration of how your implementation facilitates this process. Reassuringly the groups of conserved genes have substantive evidence to verify there involvement in aging processes. This illustrates the potential of this tool to aid experimental biologist, realising the full potential of comparative transcriptomic data analysis, informing and targeting future laboratory experimentation.

In addition to these major findings it has been extremely useful and informative to exploit your interface to provide extended annotation for mouse and nematode array reporters from there GeneBank accessions. This has allowed our research to dynamically update the annotations and reflect the highly dynamic nature of the annotation of these genomes.

Yours sincerely,

Dr Peter Kille

Reference

[1] "Genomics of Cardiovascular Development, Adaptation, and Remodeling. NHLBI Program for Genomic Applications, Harvard Medical School. URL: http://www.cardiogenomics.org," [Accessed 01/06/2006].

[2] "NC-IUB, Enzyme Nomenclature: Nomenclature Committee of the International Union of Biochemistry (NC-IUB) on the nomenclature and classification of enzymes," *Biol Chem*, 1992.

[3] I. 3rd Millennium, "Practical Data Integration in Biopharmaceutical R&D: Strategies and Technologies. A White Paper," 3rd Millennium, Inc., Waltham, MA 02451 May 2002.

[4] A. C. Siepel, A. N. Tolopko, A. D. Farmer, P. A. Steadman, F. D. Schilkey, B. Dawn Perry, and W. D. Beavis, "An integration platform for heterogeneous bioinformatics software components," *IBM Systems Journal*, vol. 40, pp. 570 - 591, 2001.

[5] AceDB, "http://www.acedb.org," [Accessed 20/12/2006].

[6] J. Adjaye, R. Herwig, D. Herrmann, W. Wruck, A. Benkahla, T. C. Brink, M. Nowak, J. W. Carnwath, C. Hultschig, H. Niemann, and H. Lehrach, "Cross-species hybridisation of human and bovine orthologous genes on high density cDNA microarrays," *BMC Genomics*, vol. 5, pp. 83, 2004.

[7] C. A. Afshari, "Perspective: microarray technology, seeing more than spots," *Endocrinology*, vol. 143, pp. 1983-9, 2002.

[8] B. Al-Daihani, A. Gray, and P. Kille, "Bioinformatics data source integration based on Semantic Relationships across species," in *Data Mining and Bioinformatics, First International Workshop, VDMB 2006, Seoul, Korea, September 11, 2006, Revised Selected Papers*, vol. 4316, *Lecture Notes in Computer Science*, M. M. Dalkilic, S. K. and, and J. Yang, Eds. Seuol, South Korea: Springer, 2006, pp. 78-93.

[9] B. Al-Daihani, A. Gray, and P. Kille, *Extracting Metadata from Biological Experimental Data*: IEEE Computer Society, 2006.

[10] B. Al-Daihani, A. Gray, and P. Kille, "Integration and Data Mining of Bioinformatics Databases(IDMBD)," Proceedings of the Sixth Informatics Workshop of Research Students, Bradford,UK, 2005, pp. 5-8.

[11] B. Al-Daihani, A. Gray, and P. Kille, "Soft Link Model(SLM) for Bioinformatics Data Source Integration," International Symposium on Health Informatics and Bioinformatics, Turkey'05, Antalya, Turkey, 2005, pp. 98-105.

[12] B. Al-Daihani, A. Gray, and P. Kille, "User preferences in Bioinformatics data source Integration and Mining," 22nd British National Conference on Databases, Sunderland,UK, 2005, pp. 82-87.

[13] F. Al-Shahrour, R. Diaz-Uriarte, and J. Dopazo, "FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes," *Bioinformatics*, vol. 20, pp. 578 - 580, 2004.

[14] J. F. Aldana, M. Rold, I. Navas, A. J. Perez, and O. Trelles, "Integrating Biological Data Sources and Data Analysis Tools through Mediators," Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004, pp. 127.

[15] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J Mol Biol*, vol. 215, pp. 403-410, 1990.

[16] R. Amato, A. Ciaramella, N. Deniskina, C. Del Mondo, D. di Bernardo, C. Donalek, G. Longo, G. Mangano, G. Miele, G. Raiconi, A. Staiano, and R. Tagliaferri, "A multi-step approach to time series analysis and gene expression clustering," *Bioinformatics*, vol. 22, pp. 589-596, 2006.

[17] A. S. Aparicio, O. L. M. Farias, and N. dos Santos, "Applying Ontologies in the Integration of Heterogeneous Relational Databases," Australasian Ontology Workshop (AOW 2005), Sydney, Australia, 2005, pp. 11-16.

[18] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium," *Nat Genet*, vol. 25, pp. 25 - 29, 2000.

[19] F. Azuaje, H. Wang, and O. Bodenreider, "Ontology-driven similarity approaches to supporting gene functional assessment," Proceedings of the ISMB 2005 SIG meeting on Bio-ontologies., 2005, pp. 9-10.

[20] F. Azuaje, H. Wang, H. Zheng, O. Bodenreider, and A. Chesneau, "Predictive Integration of Gene Ontology-Driven Similarity and Functional Interactions," Sixth IEEE International Conference On Data Mining (ICDM 2006)-Workshops (ICDM Workshops 2006), Hong Kong, 2006, pp. 114-119.

[21] M. M. Babu, "Biological Databases and Protein Sequence Analysis," [Accessed December,2005].

[22] E. H. Baehrecke, N. Dang, K. Babaria, and B. Shneiderman, "Visualization and analysis of microarray and gene ontology data with treemaps," *BMC Bioinformatics*, vol. 5, pp. 84, 2004.

[23] P. G. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens, "TAMBIS--Transparent Access to Multiple Bioinformatics Information Sources," *Proc Int Conf Intell Syst Mol Biol*, vol. 6, pp. 25-34, 1998.

[24] P. G. Baker, C. A. Goble, S. Bechhofer, N. W. Paton, R. Stevens, and A. Brass, "An ontology for bioinformatics applications," *Bioinformatics*, vol. 15, pp. 510-520, 1999.

[25] C. Barillot, H. Benali, M. Dojat, A. Gaignard, B. Gibaud, S. Kinkingnehun, J. P. Matsumoto, M. Pelegrini-Issac, E. Simon, and L. Temal, "Federating distributed and heterogeneous information sources in neuroimaging: the NeuroBase Project," *Stud Health Technol Inform*, vol. 120, pp. 3-13, 2006.

[26] C. Batini, M. Lenzerini, and S. B. Navathe, "A comparative analysis of methodologies for database schema integration," *ACM Computing Surveys (CSUR)*, vol. 18, pp. 323-364, 1986.

[27] A. D. Baxevanis and B. F. F. Ouellette, "Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins," 2rd ed. New York: John Wiley & Sons, 2001, pp. 488.

[28] Z. Ben-Miled, N. Li, Y. Liu, Y. He, E. Lynch, and O. A. Bukhres, "On the Integration of a Large Number of Life Science Web Databases," *Lecture Notes in Bioinformatics (LNBI)*, pp. 172-186, 2004.

[29] Z. Ben Milad, Y. Liu, N. Li, and O. Bukhres, "Distributed Databases," 2003.

[30] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler, "GenBank: update," *Nucleic Acids Res*, vol. 32, pp. D23-26, 2004.

[31] H. S. Bilofsky and C. Burks, "The GenBank genetic sequence data bank," *Nucleic Acids Res*, vol. 16, pp. 1861-1863, 1988.

[32] H. S. Bilofsky, C. Burks, J. W. Fickett, W. B. Goad, F. I. Lewitter, W. P. Rindone, C. D. Swindell, and C. S. Tung, "The GenBank genetic sequence databank," *Nucleic Acids Res*, vol. 14, pp. 1-4, 1986.

[33] J. A. Blake, J. T. Eppig, J. E. Richardson, C. J. Bult, and J. A. Kadin, "The Mouse Genome Database (MGD): integration nexus for the laboratory mouse," *Nucleic Acids Res*, vol. 29, pp. 91-94, 2001.

[34] J. A. Blake, J. E. Richardson, C. J. Bult, J. A. Kadin, and J. T. Eppig, "MGD: the Mouse Genome Database," *Nucleic Acids Res*, vol. 31, pp. 193-195, 2003.

[35] J. A. Blake, J. E. Richardson, C. J. Bult, J. A. Kadin, and J. T. Eppig, "The Mouse Genome Database (MGD): the model organism database for the laboratory mouse," *Nucleic Acids Res*, vol. 30, pp. 113-115, 2002.

[36] J. Bleiholder, F. Naumann, Z. Lacroix, L. Raschid, H. Murthy, and M. Vidal, "BioFast: challenges in exploring linked life sciences sources," *ACM SIGMOD Record*, vol. 33, pp. 72-77, 2004.

[37] B. Boeckmann, A. Bairoch, R. Apweiler, M. C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider, "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003," *Nucleic Acids Res*, vol. 31, pp. 365-70, 2003.

[38] C. M. Bouton and J. Pevsner, "DRAGON View: information visualization for annotated microarray data," *Bioinformatics*, vol. 18, pp. 323-324, 2002.

[39] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkinson, A. Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, J. Vilo, and M. Vingron, "Minimum information about a microarray experiment (MIAME)-toward standards for microarray data," *Nat Genet*, vol. 29, pp. 365-71, 2001.

[40] F. Bry and P. Kröger, "A Computational Biology Database Digest: Data, Data Analysis, and Data Management," *Distributed and Parallel Databases*, vol. 13, pp. 7-42, 2003.

[41] C. J. Bult, J. A. Blake, J. E. Richardson, J. A. Kadin, J. T. Eppig, R. M. Baldarelli, K. Barsanti, M. Baya, J. S. Beal, W. J. Boddy, D. W. Bradt, D. L. Burkart, N. E. Butler, J. Campbell, R. Corey, L. E. Corbani, S. Cousins, H. Dene, H. J. Drabkin, K. Frazer, D. M. Garippa, L. H. Glass, C. W. Goldsmith, P. L. Grant, B. L. King, M. Lennon-Pierce, J. Lewis, I. Lu, C. M. Lutz, L. J. Maltais, L. M. McKenzie, D. Miers, D. Modrusan, L. Ni, J. E. Ormsby, D. Qi, S. Ramachandran, T. B. Reddy, D. J. Reed, R. Sinclair, D. R. Shaw, C. L. Smith, P. Szauter, B. Taylor, P. Vanden Borre, M. Walker, L. Washburn, I. Witham, J. Winslow, and Y. Zhu, "The Mouse Genome Database (MGD): integrating biology with the genome," *Nucleic Acids Res*, vol. 32 Database issue, pp. D476-481, 2004.

[42] R. E. Buntrock, "Chemical registries--in the fourth decade of service," *J Chem Inf Comput Sci*, vol. 41, pp. 259-263, 2001.

[43] S. L. Cao, L. Qin, W. Z. He, Y. Zhong, Y. Y. Zhu, and Y. X. Li, "Semantic search among heterogeneous biological databases based on gene ontology," *Acta Biochim Biophys Sin (Shanghai)*, vol. 36, pp. 365-370, 2004.

[44] R. Carel, "Practical Data Integration In Biopharmaceutical Research and Development," *PharmaGenomics*, pp. 22-35, 2003.

[45] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom, "The TSIMMIS Project: Integration of heterogeneous information sources," Proceeedings of the Information Processing Society of Japan Conference, Tokyo, Japan, 1994, pp. 7-18.

[46] N. Chen, T. W. Harris, I. Antoshechkin, C. Bastiani, T. Bieri, D. Blasiar, K. Bradnam, P. Canaran, J. Chan, C. K. Chen, W. J. Chen, F. Cunningham, P. Davis, E. Kenny, R. Kishore, D. Lawson, R. Lee, H. M. Muller, C. Nakamura, S. Pai, P. Ozersky, A. Petcherski, A. Rogers, A. Sabo, E. M. Schwarz, K. Van Auken, Q. Wang, R. Durbin, J. Spieth, P. W. Sternberg, and L. D. Stein, "WormBase: a comprehensive data resource for Caenorhabditis biology and genomics," *Nucleic Acids Res*, vol. 33, pp. D383-389, 2005.

[47] H. J. Chung, C. H. Park, M. R. Han, S. Lee, J. H. Ohn, J. Kim, J. Kim, and J. H. Kim, "ArrayXPath II: mapping and visualizing microarray gene-expression data with biomedical ontologies and integrated biological pathway resources using Scalable Vector Graphics," *Nucleic Acids Res*, vol. 33, pp. W621-6, 2005.

[48] T. Clark, S. Martin, and T. Liefeld, "Globally distributed object identification for biological knowledgebases," *Brief Bioinform*, vol. 5, pp. 59-70, 2004.

[49] G. Cochrane, P. Aldebert, N. Althorpe, M. Andersson, W. Baker, A. Baldwin, K. Bates, S. Bhattacharyya, P. Browne, A. van den Broek, M. Castro, K. Duggan, R. Eberhardt, N. Faruque, J. Gamble, C. Kanz, T. Kulikova, C. Lee, R. Leinonen, Q. Lin, V. Lombard, R. Lopez, M. McHale, H. McWilliam, G. Mukherjee, F. Nardone, M. P. Pastor, S. Sobhany, P. Stoehr, K. Tzouvara, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler, "EMBL Nucleotide Sequence Database: developments in 2005," *Nucleic Acids Res*, vol. 34, pp. D10-15, 2006.

[50] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM*, vol. 13, pp. 377-387, 1970.

[51] C. Collet, M. N. Huhns, and W.-M. Shen, "Resource Integration Using a Large Knowledge Base in Carnot," *IEEE Computer*, vol. 24, pp. 55-62, 1991.

[52] ComparaGRID, "http://www.comparagrid.org," [Accessed 01/10/2005].

[53] K. Coyle, "Understanding Metadata and Its Purpose," *The Journal of Academic Librarianship*, vol. 31, pp. 160-163, 2005.

[54] W. Cui and H. Wu, "Using ontology to achieve the semantic integration and interoperation of GIS," Proceedings of Geoscience and Remote Sensing Symposium, 2005. IGARSS '05., 2005, pp. 836-838.

[55] N. Dao, P. J. McCormick, and C. F. Dewey, Jr., "The human physiome as an information environment," *Ann Biomed Eng*, vol. 28, pp. 1032-1042, 2000.

[56] C. J. Date, "A formal definition of the relational model," *ACM SIGMOD Record*, vol. 13, pp. 18 - 29, 1982.

[57] C. J. Date, *An introduction to Database systems*, 6 ed: Addison-Wesley publishing company, 1994.

[58] S. Davidson, C. Overton, V. Tannen, and L. Wong, "BioKleisli: A digital library for biomedical researchers," *International Journal of Digital Libraries*, vol. 1, pp. 36-53, 1997.

[59] S. B. Davidson, J. Crabtree, B. P. Brunk, J. Schug, V. Tannen, G. C. Overton, and C. J. Stoeckert, Jr., "K2/Kleisli and GUS: experiments in integrated access to genomic data sources," *IBM Systems Journal*, vol. 40, pp. 512-531, 2001.

[60] S. B. Davidson, C. Overton, and P. Buneman, "Challenges in integrating biological data sources," *J Comput Biol*, vol. 2, pp. 557-72, 1995.

[61] U. Dayal and H.-Y. Hwang, "View Definition and Generalization for Database Integration in a Multidatabase System," *IEEE Trans. Software Eng*, vol. 10, pp. 628-645, 1984.

[62] S. Decker, M. Erdmann, D. Fensel, and R. Studer, "Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information," Database Semantics - Semantic Issues in Multimedia Systems, Proceedings TC2/WG 2.6 8th Working Conference on Database Semantics (DS-8), Rotorua, New Zealand, 1999, pp. 351-369.

[63] G. Dennis, Jr., B. T. Sherman, D. A. Hosack, J. Yang, W. Gao, H. C. Lane, and R. A. Lempicki, "DAVID: Database for Annotation, Visualization, and Integrated Discovery," *Genome Biol*, vol. 4, pp. P3, 2003.

[64] D. Dinakarpandian and V. Kumar, "BIOMIND-Protein Property Prediction by Property Proximity Profiles," 17th ACM Symp. on Applied Computing (SAC) 2002., Madrid, Spain., 2002, pp. 68-72.

[65] H.-H. Do and E. Rahm, "Flexible Integration of Molecular-biological Annotation Data: The GenMapper Approach," 2003.

[66] R. M. Duwairi, "Views for Interoperability in a Heterogeneous Object-Oriented Multidatabase System," Ph.D. dissertation, Cardiff University, Cardiff, 1997.

[67] S. R. Egglestone, M. N. Alpdemir, C. Greenhalgh, A. Mukherjee, and I. Roberts, "A portal interface to myGrid workflow technology," All Hands Meeting (AHM2005), Nottingham, 2005, pp.

[68] S. R. Egglestone, M. N. Alpdemir, C. Greenhalgh, A. Mukherjee, and I. Roberts., "A Portal Interface to myGrid Workflow Technology," 2005.

[69] A. K. Elmagarmid, J. Chen, and O. A. Bukhres, "Remote System Interfaces: an Approach to Overcoming the Heterogeneity Barrier and Retaining Local Autonomy in the Integration of Heterogeneous Systems Int.," *Journal of Intelligent and Cooperative Information Systems*, vol. 2, pp. 1-22, 1993.

[70] R. Elmasri and S. B. Navathe, *Fundamentals of database systems*, 3 ed. New York: Addison-Wesley, 2000.

[71] J. T. Eppig, C. J. Bult, J. A. Kadin, J. E. Richardson, and J. A. Blake, "The Mouse Genome Database (MGD): from genes to mice--a community resource for mouse biology," *Nucleic Acids Res*, vol. 33 Database Issue, pp. D471-475, 2005.

[72] R. Erhard and A. B. Philip, "A survey of approaches to automatic schema matching," *The VLDB Journal*, vol. 10, pp. 334-350, 2001.

[73] ermineJ, "http://www.bioinformatics.ubc.ca/ermineJ/," [Accessed 27/8/2006].

[74] T. Etzold, A. Ulyanov, and P. Argos, "SRS: information retrieval system for molecular biology data banks," *Methods Enzymol*, vol. 266, pp. 114-128, 1996.

[75] R. C. C. N. H. G. F. W. Howell, "Catalyzer: a novel tool for integrating, managing and publishing heterogeneous bioscience data," *Concurrency and Computation: Practice and Experience*, vol. 19, pp. 207-221, 2007.

[76] W. M. Fitch, "Distinguishing homologous from analogous proteins," *Syst Zool*, vol. 19, pp. 99-113, 1970.

[77] A. Freier, R. Hofestadt, M. Lange, U. Scholz, and A. Stephanik, "BioDataServer: a SQL-based service for the online integration of life science data," *In Silico Biol*, vol. 2, pp. 37-57, 2002.

[78] S. Gala and W. Kim, "Database design methodology: Converting a relational schema into an object-relational schema," International Symposium on Advanced Database Technologies and Their Integration, Nara, Japan, 1994, pp. 9-33.

[79] M. Y. Galperin, "The Molecular Biology Database Collection: 2004 update," *Nucleic Acids Res*, vol. 32, pp. D3-22, 2004.

[80] M. Y. Galperin, "The Molecular Biology Database Collection: 2005 update," *Nucleic Acids Res*, vol. 33, pp. D5-24, 2005.

[81] M. Y. Galperin, "The Molecular Biology Database Collection: 2006 update," *Nucleic Acids Res*, vol. 34, pp. D3-5, 2006.

[82] M. Y. Galperin, "The Molecular Biology Database Collection: 2007 update," *Nucleic Acids Res*, vol. 35, pp. D3-4, 2007.

[83] M. Y. Galperin, "The Molecular Biology Database Collection: 2008 update," *Nucleic Acids Res*, vol. 36, pp. D2-4, 2008.

[84] M. Garcia-Solaco, F. Saltor, and M. Castellanos, "Semantic heterogeneity in multidatabase systems," in *Object-Oriented Multidatabase Systems: A Solution for Advanced Applications*, O. A. Bukhres and A. K. Elmagarmid, Eds. Hertfordshire, UK: Prentice Hall International (UK) Ltd., 1995, pp. 129 - 202.

[85] GenBank, "http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html," [Accessed 07/007/2008].

[86] GenBank, "http://www.ncbi.nlm.nih.gov/Genbank/index.html," [Accessed 26/03/2007].

[87] GEO, "http://www.ncbi.nlm.nih.gov/geo/," [Accessed 01/12/2006].

[88] D. R. Gilbert, M. Schroeder, and J. van Helden, "Interactive visualization and exploration of relationships between biological objects," *Trends Biotechnol*, vol. 18, pp. 487-494, 2000.

[89] GO, "http://www.geneontology.org," [Accessed 01/03/2005].

[90] C. A. Goble, R. Stevens, G. Ng, S. Bechhofer, N. W. Paton, P. G. Baker, M. Peim, and A. Brass, "Transparent access to multiple bioinformatics information sources," *IBM Systems Journal*, vol. 40, pp. 532-551, 2001.

[91] B. M. Good and M. D. Wilkinson, "The Life Sciences Semantic Web is Full of Creeps!" *Brief Bioinform %R 10.1093/bib/bbl025*, vol. 7, pp. 275-286, 2006.

[92] D. Greenbaum and M. Gerstein, "A universal legal framework as a prerequisite for database interoperability," in *Nature Biotechnology*: Nature Publishing Group, 2003, pp. 979 - 982.

[93] T. R. Gruber, "Towards Principles for the Design of Ontologies used for Knowledge Sharing," *International Journal of Human-Computer Studies*, vol. 43, pp. 907-928, 1995.

[94] A. Gupta, B. Ludascher, and M. Martone, "Knowledge-based integration of neuroscience data sources," 12th International Conference on Scientific and Statistical Database Management (SSDBM), Berlin, Germany, 2000, pp. 39-52.

[95] GUS, "The Genomics Unified Schema(GUS) Platform for Functional Genomics. URL: http://www.gusdb.org," [Accessed 17/04/2004].

[96] L. M. Haas, E. T. Lin, and M. A. Roth, "Data integration through database federation," *IBM Syst. J.*, vol. 41, pp. 578-596, 2002.

[97] L. M. Haas, P. M. Schwarz, P. Kodali, E. Kotlar, J. E. Rice, and W. C. Swope, "DiscoveryLink: a system for integrated access to life sciences data sources," *IBM Syst. J.*, vol. 40, pp. 489-511, 2001.

[98] J. Hammer, "Resolving Semantic Heterogeneity in a Federation of Autonomous, Heterogeneous database Systems," Ph.D. dissertation, University of Southern California, Los Angeles, CA 90089-0781, 1994.

[99] J. Hammer and D. McLeod, "An Approach to Resolving Semantic Heterogeneity in a Federation of Autonomous, Heterogeneous Database Systems," *Journal of Intelligent and Cooperative Information Systems*, vol. 2, pp. 51-83, 1993.

[100] R. C. Hardison, "Comparative genomics," *PLoS Biol*, vol. 1, pp. E58, 2003.

[101] J. Heflin and J. A. Hendler, *Dynamic Ontologies on the Web*: AAAI Press / The MIT Press, 2000.

[102] J. Heflin, J. A. Hendler, and S. Luke, "SHOE: A knowledge representation language for internet applications," Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland, Technical report 1999.

[103] M. A. Hibbs, N. C. Dirksen, K. Li, and O. G. Troyanskaya, "Visualization methods for statistical analysis of microarray clusters," *BMC Bioinformatics*, vol. 6, pp. 115, 2005.

[104] D. M. Hillis, *Homology in molecular biology. In: Homology: the hierarchical basis of comparative biology*. San Diego.: B.K. Academic Press, 1994.

[105] P. Hingamp, A. E. van den Broek, G. Stoesser, and W. Baker, "The EMBL Nucleotide Sequence Database. Contributing and Accessing data," *Mol Biotechnol*, vol. 12, pp. 255-267, 1999.

[106] D. A. Hosack, G. Dennis, Jr., B. T. Sherman, H. C. Lane, and R. A. Lempicki, "Identifying biological themes within lists of genes with EASE," *Genome Biol*, vol. 4, pp. R70, 2003.

[107] http://genomicsgtl.energy.gov/compbio/dataanalysis.shtml, "Data Analysis and Reduction," [Accessed 10/10/2006].

[108] D. Hull, R. Stevens, P. Lord, C. Wroe, and K. Goble, "Treating shimantic web syndrome with ontologies," Proceedings of First Advanced Knowledge Technologies Workshop on Semantic Web Services (AKT-SWS04), KMi, The Open University, Milton Keynes, UK, 2004, pp.

[109] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn, "Taverna: a tool for building and running workflows of services," *Nucleic Acids Res*, vol. 34, pp. W729-32, 2006.

[110] R. Hull, "Managing semantic heterogeneity in databases: A theoretical perspective," Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Tucson, Arizona, 1997, pp. 51--61.

[111] T. Hulsen, M. A. Huynen, J. de Vlieg, and P. M. Groenen, "Benchmarking ortholog identification methods using functional genomics data," *Genome Biol*, vol. 7, pp. R31, 2006.

[112] M. A. Huynen and P. Bork, "Measuring genome evolution," *Proc Natl Acad Sci U S A*, vol. 95, pp. 5849-5856, 1998.

[113] H. V. Jagadish and F. Olken, "Database management for life science research: summary report of the workshop on data management for molecular and cell biology at the National Library of Medicine, Bethesda, Maryland, February 2-3, 2003," *Omics*, vol. 7, pp. 131-7, 2003.

[114] H. V. Jagadish and F. Olken, "Database Management for Life Sciences Research," *SIGMOD Record*, vol. 33, pp. 15-20, 2004.

[115] Y. Ji, K. Coombes, J. Zhang, S. Wen, J. Mitchell, L. Pusztai, W. F. Symmans, and J. Wang, "RefSeq refinements of UniGene-based gene matching improve the correlation of expression measurements between two microarray platforms," *Appl Bioinformatics*, vol. 5, pp. 89-98, 2006.

[116] J. J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," International Conference Research on Computational Linguistics (ROCLING X), Taiwan, 1997, pp.

[117] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa, "From genomics to chemical genomics: new developments in KEGG," *Nucleic Acids Res*, vol. 34, pp. D354-357, 2006.

[118] C. Kanz, P. Aldebert, N. Althorpe, W. Baker, A. Baldwin, K. Bates, P. Browne, A. van den Broek, M. Castro, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, J. Gamble, F. G. Diez, N. Harte, T. Kulikova, Q. Lin, V. Lombard, R. Lopez, R. Mancuso, M. McHale, F. Nardone, V. Silventoinen, S. Sobhany, P. Stoehr, M. A. Tuli, K. Tzouvara, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler, "The EMBL Nucleotide Sequence Database," *Nucleic Acids Res*, vol. 33, pp. D29-33, 2005.

[119] K. A. Karasavvas, R. Baldock, and A. Burger, "Bioinformatics integration and agent technology," *J Biomed Inform*, vol. 37, pp. 205-219, 2004.

[120] S. Karlin and S. F. Altschul, "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes," *Proc Natl Acad Sci U S A*, vol. 87, pp. 2264-2268, 1990.

[121] S. Karlin, B. E. Blaisdell, and V. Brendel, "Identification of significant sequence patterns in proteins," *Methods Enzymol*, vol. 183, pp. 388-402, 1990.

[122] D. Karunaratna, "Exploitation of Semantic Information in the Creation of Multiple Views of a Federated Database Systems," Ph.D. dissertation, Cardiff University, Cardiff, 2000.

[123] V. Kashyap and A. Sheth, "Semantic and Schematic Similarities between Database Objects: A Context-based approach," *The International Journal of Very Large Data Bases (VLDB)*, vol. 5, pp. 276-304, 1996.

[124] E. Kawas, M. Senger, and M. D. Wilkinson, "BioMoby extensions to the Taverna workflow management and enactment software," *BMC Bioinformatics*, vol. 7, pp. 523, 2006.

[125] C. Kendig, "Reconstructing the Concept of Homology for Genomics," Pitt-London Workshop in the Philosophy of Biology and Neuroscience, London, 2001, pp.

[126] W. Kim, *Introduction to Object-Oriented Databases*. Cambridge, MA: The MIT Press, 1990.

[127] W. Kim, "Object-Relational database technology: a UniSQL," UniSQL Inc., Austin, TX, whitepaper 1996.

[128] W. Kim, I. Choi, S. K. Gala, and M. Scheevel, "On Resolving Schematic Heterogeneity in Multidatabase Systems," *Distributed and Parallel Databases*, vol. 1, pp. 251-279, 1993.

[129] W. Kim and J. Seo, "Classifying schematic and data heterogeneity in multidatabase systems," *IEEE Computer*, vol. 24, pp. 12-18, 1991.

[130] W. Klas and M. Schrefl, *Metaclasses and their application: data model tailoring and database integration*. Berlin; New York: Springer, 1995.

[131] J. Kohler, "Integration of life science databases," *Drug Discovery Today: BIOSILICO*, vol. 2, pp. 61-69, 2004.

[132] J. Kohler, "SEMEDA: Ontology based semantic integration of biological databases," Ph.D. dissertation, University of Bielefeld, Bielefeld, 2003.

[133] J. Kohler, J. Baumbach, J. Taubert, M. Specht, A. Skusa, A. Ruegg, C. Rawlings, P. Verrier, and S. Philippi, "Graph-based analysis and visualization of experimental results with ONDEX," *Bioinformatics*, vol. 22, pp. 1383-1390, 2006.

[134] J. Kohler, S. Philippi, and M. Lange, "SEMEDA: ontology based semantic integration of biological databases," *Bioinformatics*, vol. 19, pp. 2420-2427, 2003.

[135] J. Kohler and S. Schulze-Kremer, "The semantic metadatabase (SEMEDA): ontology based integration of federated molecular biological data sources," *In Silico Biol*, vol. 2, pp. 219-231, 2002.

[136] T. Kosaka, Y. Tohsato, S. Date, H. Matsuda, and S. Shimojo, "An OGSA-based integration of life-scientific resources for drug discovery," *Methods Inf Med*, vol. 44, pp. 257-61, 2005.

[137] T. Kulikova, P. Aldebert, N. Althorpe, W. Baker, K. Bates, P. Browne, A. van den Broek, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, M. Garcia-Pastor, N. Harte, C. Kanz, R. Leinonen, Q. Lin, V. Lombard, R. Lopez, R. Mancuso, M. McHale, F. Nardone, V. Silventoinen, P. Stoehr, G. Stoesser, M. A. Tuli, K. Tzouvara, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler, "The EMBL Nucleotide Sequence Database," *Nucleic Acids Res*, vol. 32, pp. D27-30, 2004.

[138] Z. Lacroix and T. Critchlow, *Bioinformatics: Managing Scientific Data*. San Francisco, CA: Morgan Kaufmann Publishers, 2003.

[139] Z. Lacroix, B. Omar, and E. Mehdi, *The biological integration system*. New Orleans, Louisiana, USA: ACM Press, 2003.

[140] U. Leser and F. Naumann, "(Almost) Hands-Off Information Integration for the Life Science," the Conference in Innovative Database Research (CIDR) 2005, Asilomar, Canada, 2005, pp. 131-143.

[141] A. M. Lesk, *Database Annotation in Molecular Biology: Principles and Practice*, 1 ed. West Sussex, England: Johan Wiley & Sons Inc., 2005.

[142] D. Lin, "An Information-Theoretic Definition of Similarity," Proceedings of 15th International Conference on Machine Learning, Madison, Wisconsin, 1998, pp. 296 - 304.

[143] D. J. Lipman and W. R. Pearson, "Rapid and sensitive protein similarity searches," *Science*, vol. 227, pp. 1435-1441, 1985.

[144] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble, "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation," *Bioinformatics*, vol. 19, pp. 1275-1283, 2003.

[145] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez Gene: gene-centered information at NCBI," *Nucleic Acids Res*, vol. 33, pp. D54-8, 2005.

[146] M. Maibaum, L. Zamboulis, G. Rimon, C. Orengo, N. Martin, and A. Poulovassilis, "Cluster Based Integration of Heterogeneous Biological Databases Using the AutoMed Toolkit," *Lecture Notes in Computer Science*, vol. 3615, pp. 191 - 207, 2005.

[147] P. Martin, F. Enrico, W. P. Norman, and A. G. Carole, *Query Processing with Description Logic Ontologies Over Object-Wrapped Databases*: IEEE Computer Society, 2002.

[148] S. A. McCarroll, C. T. Murphy, S. Zou, S. D. Pletcher, C. S. Chin, Y. N. Jan, C. Kenyon, C. I. Bargmann, and H. Li, "Comparing genomic expression patterns across species identifies shared transcriptional profile in aging," *Nat Genet*, vol. 36, pp. 197-204, 2004.

[149] J. McCarthy, "Metadata management for large statistical database," The Eighth International Conference on Very Large Database Systems, Mexico City, 1992, pp. 470-502.

[150] S. McCouch, "Toward a plant genomics initiative: Thoughts on the value of cross-species and cross-genera comparisons in the grasses," *PNAS*, vol. 95, pp. 1983-1985, 1998.

[151] V. A. McKusick, *Mendelian Inheritance in Man. A Catalog of Human Genes and Genetic Disorders*, 12th ed. Baltimore: Johns Hopkins University Press, 1998.

[152] E. Mehdi, B. Omar, C. Franois-Marie, and L. Yassine, "Query processing in a geographic mediation system," Proceedings of the 12th annual ACM international workshop on Geographic information systems, Washington DC, USA, 2004, pp. 101-108.

[153] P. Mitra and G. Wiederhold, "Resolving terminological heterogeneity in ontologies," Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02) workshop on Ontologies and Semantic Interoperability, Lyon, France, 2002, pp.

[154] S. B. Navathe and U. Patil, "Genomic and Proteomic Databases and Applications: A Challenge for Database Technology," in *Database Systems for Advanced Applications*, vol. 2973/2004, *Lecture Notes in Computer Science*. Heidelberg: Springer Berlin, 2004, pp. 1-24.

[155] C. B. Necib and J. C. Freytag, "Using Ontologies for Database Query Reformulation," ADBIS (Local Proceedings), 2004, pp.

[156] E. Neumann, "A Life Science Semantic Web: Are We There Yet?" *Sci. STKE*, vol. 2005, pp. pe22-, 2005.

[157] S.-K. Ng and L. Wong, "Accomplishments and Challenges in Bioinformatics," *IT Professional*, vol. 6, pp. 40 - 50, 2004.

[158] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, "Taverna: a tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, vol. 20, pp. 3045-54, 2004.

[159] Ostell, JM., S. J. Wheelan, and J. A. Kans, *The NCBI Data Model in Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, 2 ed: John Wiley & Sons Publishing, 2001.

[160] OWL, "http://www.w3.org/TR/owl-ref/," [Accessed 28/07/2008].

[161] M. Palakal, S. Mukhopadhyay, and M. Stephens, "Identification of Biological Relationships from Text Documents," in *Medical Informatics*, vol. 8, *Integrated Series in Information Systems*: Springer US, 2006, pp. 449-489.

[162] C. Partridge, "The Role of Ontology in Integrating Semantically Heterogeneous Databases," Padova June 2002.

[163] C. Pasquier, "Biological data integration using Semantic Web technologies," *Biochimie*, vol. 90, pp. 584-94, 2008.

[164] C. Patterson, "Homology in classical and molecular biology," *Mol Biol Evol*, vol. 5, pp. 603-625, 1988.

[165] S. Philippi and J. Kohler, "Using XML technology for the ontology-based semantic integration of life science databases," *IEEE Trans Inf Technol Biomed*, vol. 8, pp. 154-60, 2004.

[166] K. D. Pruitt and D. R. Maglott, "RefSeq and LocusLink: NCBI gene-centered resources," *Nucleic Acids Res*, vol. 29, pp. 137-140, 2001.

[167] C. Raguenaud, "Managing complex taxonomic data in an object-oriented database," PhD, Napier University, Edinburgh, 2002.

[168] RDF, "http://www.w3.org/TR/rdf-concepts/," [Accessed 28/07/2008].

[169] A. L. Rector, S. Bechhofer, C. A. Goble, I. Horrocks, W. A. Nowlan, and W. D. Solomon, "The GRAIL concept modelling language for medical terminology," *Artificial Intelligence in Medicine*, vol. 9, pp. 139-171, 1997.

[170] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, 1995, pp. 448-453.

[171] D. Reynolds, C. Thompson, J. Mukerji, and D. Coleman, "An assessment of RDF/OWL modelling," Hewlett Packard Lab, Technical Report 2005-189 October, 28 2005.

[172] P. A. Rioux, W. A. Gilbert, and T. G. Littlejohn, "A portable search engine and browser for the Entrez database," *J Comput Biol*, vol. 1, pp. 293-295, 1994.

[173] H. Robert. and M. Patricia., "SRS as a possible infrastructure for GBIF. GBIF DADI Meeting," San Diego June 26-28 2002 2002.

[174] I. Rojas, E. Ratsch, J. Saric, and U. Wittig, "Notes on the use of ontologies in the biochemical domain," *In Silico Biol*, vol. 4, pp. 89-96, 2004.

[175] G. D. Schuler, J. A. Epstein, H. Ohkawa, and J. A. Kans, "Entrez: molecular biology database and retrieval system," *Methods Enzymol*, vol. 266, pp. 141-162, 1996.

[176] S. Schulze-Kremer, "Ontologies for molecular biology and bioinformatics," *In Silico Biol*, vol. 2, pp. 179-193, 2002.

[177] S. Schweigert, P. V. Herde, and P. R. Sibbald, "Issues in incorporation semantic integrity in molecular biological object-oriented databases," *Computer Applications in the Biosciences*, vol. 11, pp. 339-347, 1995.

[178] R. Sealfon, M. Hibbs, C. Huttenhower, C. Myers, and O. Troyanskaya, "GOLEM: an interactive graph-based gene-ontology navigation and analysis tool," *BMC Bioinformatics*, vol. 7, pp. 443, 2006.

[179] A. P. Sheth and J. A. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM Computing Surveys*, vol. 22, pp. 183-236, 1990.

[180] M. Siegael and S. Madnick, "A Metadata Approach to Resolving Semantic Conflicts," Proceedings of the 17th International Conference on Very Large Data Bases, Barcelona, 1991, pp. 133-145.

[181] C. H. I. Staff and N. Goodman, "New Tools and Approaches Revolutionizing Microarray Data Analysis," 2001.

[182] R. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N. W. Paton, C. A. Goble, and A. Brass, "TAMBIS: transparent access to multiple bioinformatics information sources," *Bioinformatics*, vol. 16, pp. 184-185, 2000.

[183] R. Stevens, A. Robinson, and C. A. Goble, "myGrid: Personalised Bioinformatics on the Information Grid,"

Proceedings of 11th International Conference on Intelligent Systems in Molecular Biology, Brisbane, Australia, 2003, pp.

[184] M. Stonebraker and G. Kemnitz, "The POSTGRES next generation database management system," *Communications of the ACM*, vol. 34, pp. 78-92, 1991.

[185] M. Stonebraker and D. Moore, *Object Relational DBMSs: The Next Great Wave*. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1995.

[186] A. Sudeshna, S. B. Vishal, N. B. Deo, P. V. Kamesam, K. Pankaj, P. K. Manish, and S. Biplav, *A system for knowledge management in bioinformatics*. McLean, Virginia, USA: ACM, 2002.

[187] Y. Tateno, S. Miyazaki, M. Ota, H. Sugawara, and T. Gojobori, "DNA Data Bank of Japan (DDBJ) in collaboration with mass sequencing teams," *Nucl. Acids Res*, vol. 28, pp. 24-26, 2000.

[188] A.-R. Tawil, "Supporting Semantic Interoperability in a Heterogeneous Multiple Information Server Environment," Ph.D. dissertation, Cardiff University, Cardiff, 2001.

[189] H. Thomas and K. Subbarao, "Integration of biological sources: current systems and challenges ahead," *SIGMOD Rec.*, vol. 33, pp. 51-60, 2004.

[190] O. Tom, G. Mark, A. Matthew, M. N. Alpdemir, F. Justin, G. Kevin, G. Carole, G. Antoon, H. Duncan, M. Darren, L. Peter, L. Phillip, R. P. Matthew, S. Martin, S. Robert, W. Anil, and W. Chris, "Taverna: lessons in creating a workflow environment for the life sciences: Research Articles," *Concurr. Comput.: Pract. Exper.*, vol. 18, pp. 1067-1100, 2006.

[191] T. Topaloglou, S. B. Davidson, H. V. Jagadish, V. M. Markowitz, E. W. Steeg, and M. Tyers, "Biological Data Management: Research, Practice and Opportunities," Proceedings of the 30th VLDB Conference, Toronto, Canda, 2004, pp. 1233-1236.

[192] T. J. Troup, "A component System Architecture to Enable User Directed Component Binding at Run-Time," Ph.D. dissertation, University of Glasgow, Glasgow, 2004.

[193] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, "A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae)," *Proc Natl Acad Sci U S A*, vol. 100, pp. 8348-8353, 2003.

[194] T. V. Venkatesh and H. B. Harlow, "Integromics: challenges in data integration," *Genome Biology*, vol. 3, pp. REPORTS4027.1 - REPORTS4027.3, 2002.

[195] H. Vyas and R. Summers, "Interoperability of bioinformatics resources," *VINE*, vol. 35, pp. 132-139, 2005.

[196] S. O. D. W. John MacMullen, "Information problems in molecular biology and bioinformatics," *Journal of the American Society for Information Science and Technology*, vol. 56, pp. 447-456, 2005.

[197] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hubner, "Ontology-based integration of information - a survey of existing approaches," In Proceedings of the Workshop Ontologies and Information Sharing, IJCAI, 2001, Seattle, WA, 2001, pp. 108-117.

[198] H. Wang, F. Azuaje, and O. Bodenreider, "An Ontology-Driven Clustering Method for Supporting Gene Expression Analysis," *Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium*, pp. 389- 394, 2005.

[199] J. T. L. Wang, M. J. Zaki, H. T. T. Toivonen, and D. E. Shasha, *Data Mining in Bioinformatics*. London: Springer, 2005.

[200] X. Wang, H. He, L. Li, R. Chen, X. W. Deng, and S. Li, "NMPP: a user-customized NimbleGen microarray data processing pipeline," *Bioinformatics*, vol. 22, pp. 2955-2957, 2006.

[201] G. Wiederhold, "Mediators in the Architecture of Future Information Systems," *IEEE Computer*, vol. 23, pp. 38-49, 1992.

[202] G. Wiederhold, "Objects and domains for managing medical data and knowledge," *Methods Inf Med*, vol. 34, pp. 40-46, 1995.

[203] M. Wilkinson, "Gbrowse Moby: a Web-based browser for BioMoby Services," *Source Code for Biology and Medicine*, vol. 1, pp. 4, 2006.

[204] M. Wilkinson, D. Gessler, A. Farmer, and L. Stein, "The BioMOBY project explores open-source, simple, extensible protocols for enabling biological database interoperability.," Proceedings of the Virtual Conference on Genomics and Bioinformatics, 2003, pp. 16-26.

[205] M. Wilkinson, H. Schoof, R. Ernst, and D. Haase, "BioMOBY successfully integrates distributed heterogeneous bioinformatics Web Services. The PlaNet exemplar case," *Plant Physiol*, vol. 138, pp. 5-17, 2005.

[206] M. D. Wilkinson and M. Links, "BioMOBY: an open source biological web services proposal," *Brief Bioinform*, vol. 3, pp. 331-41, 2002.

[207] M. D. Wilkinson, M. Senger, E. Kawas, R. Bruskiewich, J. Gouzy, C. Noirot, P. Bardou, A. Ng, D. Haase, A. Saiz Ede, D. Wang, F. Gibbons, P. M. Gordon, C. W. Sensen, J. M. Carrasco, J. M. Fernandez, L. Shen, M. Links, M. Ng, N. Opushneva, P. B. Neerincx, J. A. Leunissen, R. Ernst, S. Twigger, B. Usadel, B. Good, Y. Wong, L. Stein, W. Crosby, J. Karlsson, R. Royo, I. Parraga, S. Ramirez, J. L. Gelpi, O. Trelles, D. G. Pisano, N. Jimenez, A. Kerhornou, R. Rosset, L. Zamacola, J. Tarraga, J. Huerta-Cepas, J. M. Carazo, J. Dopazo, R. Guigo, A. Navarro, M. Orozco, A. Valencia, M. G. Claros, A. J. Perez, J. Aldana, M. M. Rojano, R. Fernandez-Santa Cruz, I. Navas, G. Schiltz, A. Farmer, D. Gessler, H. Schoof, and A. Groscurth, "Interoperability with Moby 1.0--it's better than sharing your toothbrush!" *Brief Bioinform*, vol. 9, pp. 220-31, 2008.

[208] L. Wong, *The Collection Programming Language Reference Manual*. Singapore: Institute of Systems Science, 1995.

[209] J. C. Wooley and H. S. Lin, "Catalyzing Inquiry at the Interface of Computing and Biology." Washington, D.C: The National Academies Press, 2005, pp. 443.

[210] WormBase, "http://www.wormbase.org/," [Accessed 01/04/2004].

[211] G. Wrobel, F. Chalmel, and M. Primig, "goCluster integrates statistical analysis and functional interpretation of microarray expression data," *Bioinformatics*, vol. 21, pp. 3575-3577, 2005.

[212] Z. Wu and M. Palmer, "Verb semantics and lexical selection," Proceedings of the 32nd Annual Meetings of the Associations for Computational Linguistics, Las Cruces, New Mexico, 1994, pp. 133-138.

[213] Q. Xu, Y. Shi, Q. Lu, G. Zhang, Q. Luo, and Y. Li, "GORouter: an RDF model for providing semantic query and inference services for Gene Ontology and its associations," *BMC Bioinformatics*, vol. 9, pp. S6, 2008.

[214] S. Zhang, "ExperiBase: an integrated software architecture to support modern experimental biology," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.