INSTITUT FÜR INFORMATIK
Lehr- und Forschungseinheit für
Programmier- und Modellierungssprachen

Oettingenstraße 67 D–80538 München

Ludwig
Maximilians-
Universität
München

Molecular Biology Data : Database Overview, Modelling Issues, and Perspectives

Peer Kröger

Diplomarbeit

Beginn der Arbeit: 18.12.2000
Abgabe der Arbeit: 18.06.2001
Betreuer: Prof. Dr. François Bry
PD Dr. Rolf Backofen

Erklärung

Hiermit versichere ich, dass ich diese Diplomarbeit selbständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Zorneding, den 17.06.2001 Peer Kröger

Abstract

Modern methods in molecular biology produce a tremendous amount of data. These biological data have to be stored adequately for both data retrieval and data analysis. This is important for biology which is a knowledge-based and data-intensive discipline.

In fact, the biology community is a distributed one. Most research teams generate their own data and often build or maintain their own electronic resources. In the past few years the number of molecular biology databases accessible via WWW grew steadily and the importance of these databases increases constantly. Often newly produced data are no longer published in scientific journals but instead are accessible only from these databases. The available flood of data can no longer be managed by the brains of human domain experts.

For computer scientists, it is in general difficult to get an overview over these molecular biology databases, their special needs, and their evolution. This situation is unfortunate because molecular biology databases are very interesting not only for biologists but also for computer scientists. Indeed, they pose challenging warehousing and knowledge representation problems. Computer scientists attempting to understand this domain of modern molecular biology or trying to keep track of these databases often give up confused and frustrated. The main reason for this is that there is no basic introduction into molecular biology databases and their particularities providing an overview with literature references and links to databases. This work tries to fill this lack.

The primary objective of this thesis is to give an up-to-date overview of the (probably) most important molecular biology databases. Of course the extremely rapid evolution of the domain will make this overview in some of its aspects outdated after a few years or even months. However a "snapshot" as of Spring 2001 is likely to remain useful. Links to and bibliographic references with more detailed information about each database considered in this overview are provided.

A second objective of this thesis is to introduce into knowledge representation for molecular biology. This is an emerging area of research which already resulted in original approaches. In this work it is distinguished between (primary) symbolic and (primary) numerical approaches (i.e. mathematical approaches such as differential equations). The focus is on symbolic knowledge representation.

This work has been written for a readership of computer scientists. Therefore it also provides a short and hopefully easy understandable introduction into the essentials of molecular biology. Finally the integration of heterogeneous molecular biology databases and the modelling of regulatory and metabolic pathways are briefly reviewed. They are emerging issues of considerable practical relevance.

Zusammenfassung

Moderne Methoden der Molekularbiologie produzieren eine unvorstellbar große Menge an Daten. Diese biologische Daten müssen angemessen verwaltet und gespeichert werden, um das Wiederauffinden und die Analyse zu ermöglichen. Dies ist besonders wichtig für die Biologie als wissensbasierte und datenintensive Disziplin.

Die meisten Forschungsgruppen der Biologie erzeugen ihre eigenen Daten und entwickeln oder verwalten im Zuge dessen auch ihre eigenen elektronischen Hilfsmittel. Im Laufe der letzten Jahre nahm die Zahl der Biologie-Datenbanken, die über das Internet erreichbar sind, konstant zu. Die Bedeutung dieser Datenbanken wächst stetig: Neu erzeugte Daten werden oft nicht mehr in wissenschaftlichen Veröffentlichungen und Zeitschriften publiziert, sondern sind stattdessen nur noch über diese Datenbanken zugänglich. Dazu kommt, dass die verfügbare Datenflut nicht länger ohne computergestützte Hilfsmittel zu bewältigen ist.

Für Informatiker ist es grundsätzlich schwierig, sich einen Überblick über diese Biologie-Datenbanken, ihre speziellen Bedürfnisse und ihre Entwicklungen zu verschaffen. Dabei ist das Gebiet der Biologie-Datenbanken nicht nur für den Biologen sehr interessant sondern auch für den Informatiker. In der Tat sind anspruchsvolle Probleme in den Bereichen *Data warehousing* und Wissensrepresentation zu lösen. Doch Informatiker, die versuchen, sich einen Überblick über dieses moderne Feld der Molekularbiologie zu verschaffen, geben oft schon bald frustriert und verwirrt auf. Dies liegt v. a. daran, dass es keine grundlegende Einführung in das Gebiet der Biologie-Datenbanken und deren Besonderheiten gibt, die einen Überblick mit Literaturverweisen und Links zu Datenbanken anbietet. Die Diplomarbeit versucht, diese Lücke zu schließen.

Das Hauptziel dieser Diplomarbeit ist es, einen aktuellen Überblick über die wichtigsten Biologie-Datenbanken zu erstellen. Natürlich wird die extrem schnelle Entwicklung in diesem Bereich der Wissenschaft ein Übriges tun, so dass einige Aspekte der Arbeit bald überholt sein werden. Dennoch wird eine solche Momentaufnahme aus dem Frühjahr 2001 höchst wahrscheinlich auch langfristig von Nutzen sein. Es werden Links und Literaturangaben, die zu detaillierteren Informationen über jede Datenbank in diesem Überblick führen, angeboten.

Ein zweites Ziel dieser Diplomarbeit ist, in das Gebiet der biologischen Wissensrepresentation einzuführen. Dieses Forschungsgebiet ist ein aufstrebender Bereich, der bereits einige interessante Ansätze produziert hat. In dieser Arbeit wurde zwischen (in erster Linie) symbolischen und nummerischen (hauptsächlich mathematischen Ansätzen wie z.B. Differentialgleichungen) Ansätzen unterschieden. Der Schwerpunkt liegt dabei auf den symbolischen Ansätzen der Wissensrepräsentation.

Diese Diplomarbeit ist für Informatiker geschrieben worden. Daher bietet sie auch eine kurze und hoffentlich leicht verständliche Einführung in die wesentlichen Begriffe der Molekularbiologie. Des Weiteren werden die Gebiete Integration von heterogenen Biologie-Datenbanken und Modellierung Regulatorischer und Metabolischer Pfade angesprochen. Beide Bereiche sind von beachtlicher praktischer Bedeutung.

Contents Overview

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biology is a knowledge-based, data intensive discipline. Biologists interprete newly collected data relying upon comparisons with formerly gathered data. Predictions are often based upon comparisons of huge collections of data. Since the knowledge of past experience plays a key role, this knowledge has to be kept in such a manner that it can be used in the most effective way [7].

Traditionally, up till the beginning or the mid of the eighties, biology data has resided mostly within the brain of experts. This pattern worked well as long as the amount of data was not overwhelming the single expert.

Now, the situation has dramatically changed. Not only is the rate of data acquisition growing exponentially – especially because of the work in so-called "sequencing" (cf. Chapter 2.6.2) – but also a single experiment can now yield a tremendous amount of data that would require an army of domain experts to interpret [7].

The amount of the already collected data worldwide is far too large for any human brain, even for any team of researchers, to gather or to have an overview of. Thus, with traditional approaches, predictions and interpretations are only possible against a very small subset of the available data. Important knowledge would not be discovered if only traditional approaches were used.

Thus, there was a need to develop systems to help the experts to accumulate and interpret the data collected. During the last ten to fifteen years, many databases have been developed for gathering molecular biology data.

It is difficult to get an accurate, up-to-date overview of molecular biology databases. Molecular biology databases are extremely numerous – one might estimate their number between 500 and 1,000. It is worth mentioning that a sort of "meta-database", called DBcat (cf. Appendix A.19), has been developed for keeping track of molecular biology databases. The number of molecular biology databases is rapidly growing and the amount of data collected in most database is exponentially growing (cf. Figure 1.1). A reason for this growth is that most molecular biology databases are very heterogeneous in their aims, shapes, and in the usages they have been developed for. This derives from the heterogeneity of biological data. There are data on hundreds of different organisms, as well as on different biological concepts. While some molecular biology databases contain only the data gathered on one specific organism (e.g. the Human Genome Database GDB (cf. Appendix A.35) on the Human Genome Project or the MIPS/Saccharomyces (cf. Appendix A.63) database on yeast), or are developed and maintained by only one research team, other molecular biology databases aim at collecting all data available on biologically interesting concepts (such as SWISS-PROT (cf. Appendix A.103), a database containing information about proteins from all organisms or GenBank (cf. Appendix A.36), a database of all publicly available nucleotide sequences) and are the result of long lasting international cooperations between research laboratories. Another fact that makes it difficult to keep track of molecular biology databases is that the most different approaches are used in molecular biology databases for data modelling, for storing, and for data analysis and query purposes.

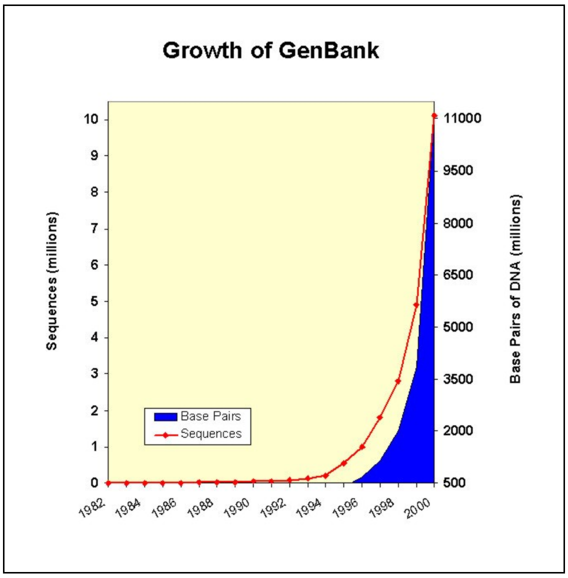


FIGURE 1.1: Example of the exponential increase of the amount of biological data: Growth-statistics of GenBank, one of the major nucleotide sequence databases [27].

Although standard data modelling and storing techniques, e.g. database systems, developed for other application areas are in use with molecular biology databases, specific approaches and tools are obviously required. The border line between necessary particular approaches and tools and the use of generic techniques seems to be rather unclear.

Furthermore, the area of molecular biology databases is a quite unstable, rapidly evolving field.

In molecular biology databases, the problems resulting from the constant, exponential growth of data are only partly overcome. On one hand, molecular biology databases often do not represent the state-of-art in database technology. Many molecular biology databases with extremely valuable contents are just collections of files – e.g. ASCII files and even GIF files! The so-called “flat files” are the *de facto* data interchange standard in the field. On the other hand, existing, up-to-date concepts of database technology are not always well-suited to biological data. Existing database techniques have to be adapted and new techniques have to be developed so as to create fully adequate information systems for molecular biology.

An acute problem that biologists querying molecular biology databases have to cope with is the heterogeneity of the existing databases. The same denomination might denote in different molecular biology databases rather different data. Possible inconsistencies are, up till now, rarely mentioned and very difficult to detect. In molecular biology databases there is a considerable need for advanced, knowledge-based data integration methods.

Furthermore, biologists often have to query several stand-alone data sources to get satisfying results. The dedicated analysis tools in use require different input-formats and are often very complex but only poorly documented. There are almost no transparent, standardized access tools to the major data sources and analysis methods currently in use in molecular biology. Thus, there is a considerable need for computational infrastructures that would help in integrating data from distinct, heterogeneous sources, and provide with sophisticated analysis tools through a transparent, unifying interface.

The cooperation of general computer scientists, not only bioinformaticians, and biologists therefore seem to be necessary for the development of such information systems. Unfortunately, not only molecular biology concepts are rather unknown to most computer scientists but also the data, databases, and data processing methods biologists and bioinformaticians are using. The techniques and tools used in molecular biology databases are unknown to most database researchers or practitioners. A further obstacle for computer scientists is that most of the literature about molecular biology databases is written by biologists for biologists. There is no basic starting point for a computer science generalist to get insights into the domain.

The progress of biological methods to generate data such as genome sequencing has been quite significant in the last two decades. But the gap between the complete genome sequence of an organism and the complete understanding of the biological function of the organism is still huge [17]. Molecular biology databases provide a huge amount of empirical facts about many different aspects of biological entities. But these data are static in the sense that molecular biology databases cannot answer questions like “How, if all, do gene A and gene B interact?” or “What effect has protein A on gene B?”. Those questions cannot be answered universally. To answer those questions requires an explicit model of the organism of interest [33].

In fact, there is a tremendous amount of data available even on interactions between single instances of biological entities. But computer-based simulation, computation and prediction is needed to understand complex systems of many biological interactions and would greatly benefit the investigations of relationships in molecular biology. This would be important since biology is a science of relationships, meaning that biologists are interested in the interactions between biological entities.

Currently, emerging research issues are concerned with the modelling of biological interactions and relationships. Well-known examples are modelling of protein docking, protein folding, regulation of gene expression (also referred to as “regulatory pathways”), and metabolic pathways. The latter two are new, rapidly evolving research areas of considerable practical relevance: they play a key role in drug design and gene therapy. Current approaches in modelling regulatory and metabolic pathways range from static representations of pathway maps (e.g. in the KEGG (cf. Appendix A.52) database) to dynamic simulations of entire cells (e.g. the E-CELL Project [68]). These rising areas of research would also greatly profit by more computer science experts concentrating on it. But solid research and development requires detailed knowledge about the biological basics. It is hard for computer science generalists to get insight into modelling of regulatory and metabolic pathways because of the same reasons that mentioned above in the context of molecular biology databases.

This thesis aims at providing with a starting point for computer science generalists to get insight into molecular biology especially into molecular biology databases. In addition, this thesis briefly introduces to and overviews current approaches in modelling regulatory and metabolic pathways.

Chapter 2 introduces to the essentials of molecular biology. This part is hopefully easy to understand and describes only few but probably the most important concepts of general molecular biology.

Chapter 3 introduces into the field of molecular biology databases focusing on the following issues:

1. molecular biology data modelling,
2. molecular biology data acquisition, and
3. specific data retrieval techniques

applied with molecular biology data. This thesis also proposes a systematic classification of a large number (120) of molecular biology databases based upon computer science criteria such as:

1. data model and storage structures,

2. data acquisition, and

3. query answering and data retrieval.

The classification criteria are described in Chapter 3. The database directory of 120 selected molecular biology databases is provided in Appendix A.

Chapter 4 proposes requirements and investigates how a transparent, unifying interface to distinct, heterogeneous data sources and advanced analysis tools could be conceived. In addition, Chapter 4 introduces in this topic and gives an overview of some approaches in the field.

Chapter 5 introduces into biological aspects of gene expression and metabolic pathways and describe current approaches to model the regulation of gene expression and metabolic pathways.

It should be mentioned, that there is a book edited by Stanley Letovsky [46] which has similar scope as this diploma thesis. It consists of articles by different authors describing fifteen of the most frequently used data repositories in molecular biology and also covers the topic of molecular biology database integration. The book is a collection of articles written by the developers of each system or approach, thus providing with many technical details.

Our thesis differs to Letovsky's book in several particularities:

First of all, we do not provide so much details on the single systems but cover a far broader spectrum of databases to be described (15 data repositories in [46] and 120 databases here). We try to provide a "snapshot" of the field that is much more comprehensive than Letovsky's book. The diploma thesis also summerizes and classifies the current state of the art in molecular biology databases and molecular biology database integration. We compare the databases and several current integration approaches according to the classification criteria described in Chapter 3. No classification and no comparison are given in [46].

A great handicap for computer scientists is that – similar to nearly all bioinformatics literature – the articles in Letovsky's book use many technical terms of biology without introducing them. Most computer scientists do not provide profound expertise in biology to understand these terms. In contrast, we try to provide with an introduction to molecular biology in order to make the biological terms used later beeing understandable.

Nevertheless, [46] is an excellent source to get an accurate and more detailed overview of a few important database systems in molecular biology. The authors of the articles are currently leading bioinformatics practioners. In our opinion, this book covers nearly all the main technologies in today's research field of molecular biology databases.

Note that [46] does not cover the modelling of biochemical pathways.

To finish this introduction, I want to mention that several persons contributed to this work.

I want to express my warmest thanks to **Prof. Dr. François Bry** for supervising my thesis and giving me the opportunity to start research in bioinformatics. I am also very grateful to **PD Dr. Rolf Backofen** for co-supervision.

I further acknowledge the funding of the **Bayerisch-Französisches-Hochschul-Zentrum (BFHZ) – Centre de Coopération Universitaire Franco-Bavarois (CCUFB)**[1] that financially supported my research and gave me the opportunity to visit the bioinformatics group of INRIA in Grenoble, France.

Special thanks are dedicated to **Dr. François Rechenmann** (INRIA Rhône-Alpes) and his team for receiving me twice at INRIA in Grenoble as well as for many fruitful discussions and for sharing great insights into the bioinformatics domain.

In addition, I am also very greatful to **Prof. Dr. Stefan Conrad** (Institute für Informatik, Ludwig-Maximilians-Universität München) for giving hints, and for reviewing and discussing Chapter 4 as well as **Dr. Johannes Herrmann** (Adolf-Butenandt-Institut für Physiologische Chemie, Ludwig-Maximilians-Universität München) for giving usefull suggestions, reviewing, and discussing Chapter 2.

Finally, I want to express my thanks to **Prof. Dr. Hans-Werner Mewes** (Munich Information Center for Protein Structures, Martinsried) for answering my questions.

[1] `http://www.bfhz.uni-muenchen.de/`

Chapter 2

A Gentle Introduction to Molecular Biology

The domain of biochemistry in general and of molecular biology in particular is concerned with the basic molecular principles of life. Biological objects interact with each others what makes possible all the different forms of life. Molecular biology focuses on these interactions which can be compared to relationships in an entity relationship model.

Modern methods in biology produce a tremendous amount of data. These biological data have to be stored adequately for both analyzing and warehousing. Dedicated computer systems are developed for these tasks.

It is no secret that computer scientists usually do not have profound expertise in the domain of molecular biology. But computer scientists need some detailed knowledge in the domain of molecular biology in order to accurately address the requirements of biologists. Unfortunately, there is no common level of abstraction for the communication between computer scientists and biologists.

This chapter is intended to computer scientists who have no or only little knowledge of molecular biology. We want to provide those persons with a minimum of information on molecular biology, paying a special attention to issues relevant for molecular biology databases. Some fundamental concepts of biochemistry are introduced. A section about biochemical and bioinformatic applications rounds up this chapter. Some methods and programs – which are closely related to molecular biology databases – are briefly addressed.

2.1 Getting Started

The domain of biochemistry in general and of molecular biology in particular are concerned with the basic molecular principles of life. They are subareas of biology. Molecular biology investigates these issues on a molecular level (a molecule is a set of atoms that are connected by chemical bonds). Molecular biology thus studies the molecules that occur in a living organism – often also called bio-molecules – and investigates their structures and functions. Note that the structure of a bio-molecule determines its function. Molecular biology also focuses on the flow of the genetic information – from genes to proteins – and how proteins affect the reactions within the organisms (details later).

Living organisms all rely on the same physical laws. These laws determine how cells – the functional organization units of all organisms – create complex structures out of simple bricks, the molecules.

The molecules and complex structures can be seen as the objects in an Entity-Relationship-Model. The relationships in this model could be the interactions between all the molecules. It should be emphasized that molecular biology as well as biology in general is a science of these relationships. Unfortunately it is not a trivial task to understand the concepts of the objects (entities) in biology. But this is necessary because the main goal of biological research is to investigate the interactions of the bio-molecules.

The computer scientist may now ask a question: ”Why are all these things of concern for computer science?”

In order to answer this question the problem mentioned above can be simplified and expressed in terms of an entity-relationship model (ER model). The complete model of life on a molecular level is not yet known. There is a tremendous amount of information on single instances, on relationships between instances and often on whole entities and their relationships. But the final model is far out of reach. Of course it is even not sure if this final model would ever be determined but parts of it would help researchers to fill the gaps because – as it has often been observed – the interactions between the ”entities” follow common laws.

In fact the amount of (biological) data which is produced to date and could be of relevance for the ER model is far too large to enable researchers to deal efficiently with it. This is the point where computer science comes into play. An enormous flood of data has to be stored in a well structured manner so that analyzing and updating of the data is conveniently supported. This requirement seems to be very familiar to computer scientists dealing with software and database development.

But to address this demand is not as easy as it may seem. In fact, computer scientists need to have detailed knowledge of the domain of biochemistry to create models and to develop analysis tools for biological data. One reason for this, is a specificity of biology: Exceptions of common (biological) laws and (biological) models may appear at any place. In fact, for most biological “theorems” there are exceptions. These exceptions cause big difficulties in building models for molecular biology.

As many computer scientists do not have profound expertise in this domain the research field in between molecular biology and computer science – called bioinformatics or computational biology – is reservated for few researchers who have detailed skills in both domains.

In the past few years the bioinformatics domain has much gained in importance. It has several sub-areas of research. The most important tasks may be the modeling of biological data and the development of specialized analysis tools. As many tools and resources have been developed for special and non-related formats, there are currently great efforts to integrate these resources and develop simple interfaces for them.

To enable computer scientists to get some insight into the domain of bioinformatics we first want to provide them with some basic knowledge on molecular biology. Our introduction is dedicated for computer scientists who do not have detailed knowledge in the domain of molecular biology. First, the flow of genetic information is overviewed. This describes the overall context of the following sections. We will then describe some properties of the objects – the bio-molecules. Some important bio-molecules will be mentioned. A more detailed discussion of the genetic information flow which is a set of procedures that enables organisms to live, grow and reproduce themselves will follow. These procedures are the basis of all known ways of life. To conclude this part and to prepare the innocent reader for the rest of the work we will have a short look at some methods, applications and tools currently used in molecular biology and bioinformatics. At the beginning of each section a list of terms of the most important biological concepts will be given.

This introduction is structured as follows:

- 2.1 Getting Started
- 2.2 The Genetic Information Flow (Overview)
- 2.3 Some Important Macro Molecules
  - 2.3.1 Nucleic Acids (DNA/RNA)
  - 2.3.2 Proteins
  - 2.3.3 Polysaccharides
  - 2.3.4 Lipids
  - 2.3.5 Summary
- 2.4 Supra-Molecular Complexes, Cells, and Cell Parasits
  - 2.4.1 Supra-Molecular Complexes
  - 2.4.2 Cells
  - 2.4.3 Cell Parasits
  - 2.4.4 Summary
- 2.5 The Genetic Information Flow (More Detailed)
  - 2.5.1 Replication
  - 2.5.2 Transcription
  - 2.5.3 Translation
  - 2.5.4 Summary
- 2.6 A Selection of Biology and Bioinformatics Applications
  - 2.6.1 Model Organisms
  - 2.6.2 Essential Methods for Generating Biological Data
    - DNA-Cloning
    - DNA-Sequencing
  - 2.6.3 Typical Bioinformatics Applications
    - Sequence Analysis
    - Protein Folding and 3D Structure Analysis

2.2 The Genetic Information Flow (Overview)

A central interest of molecular biology is the flow of information within the organisms. We will give a short overview of this information flow in this section. Sections 2.3 and 2.4 will introduce basic biochemistry concepts based on which a more detailed description of the procedures that carry out this information flow is presented in Section 2.5.

Living organisms store all information that is necessary for growth, reproduction, and evolution in so-called genes on the DNA. The genes consist of a linear sequence of so-called nucleotides. For there are only four different nucleotides appearing in DNA, a gene can be seen as a linear sequence over a four-letter alphabet.

Dedicated procedures translate the genes into so-called proteins. Proteins are linear sequences of so-called amino acids. There are 20 different amino acids that appear in proteins of living organisms, thus a protein can be modeled as a linear sequence over a 20-letter alphabet. In fact, a triplet of three nucleotides encode one amino acid. Proteins supervise all chemical reactions within the organisms. Usually, no reaction can be performed in the absence of a protein which mediates the reaction. Proteins thus determine which substances can be built out of small substances.

Figure 2.1 presents a graphical overview of the genetic information flow in living organisms. Summarizing, the genes of an organism determine which proteins can be produced by the organism and thus determine which chemical reactions can be performed in the organism. The chemical reactions enabled by the proteins encoded by the genome (complete amount of all genes) of the organism are essential for the organism's growth, evolution, and reproduction.

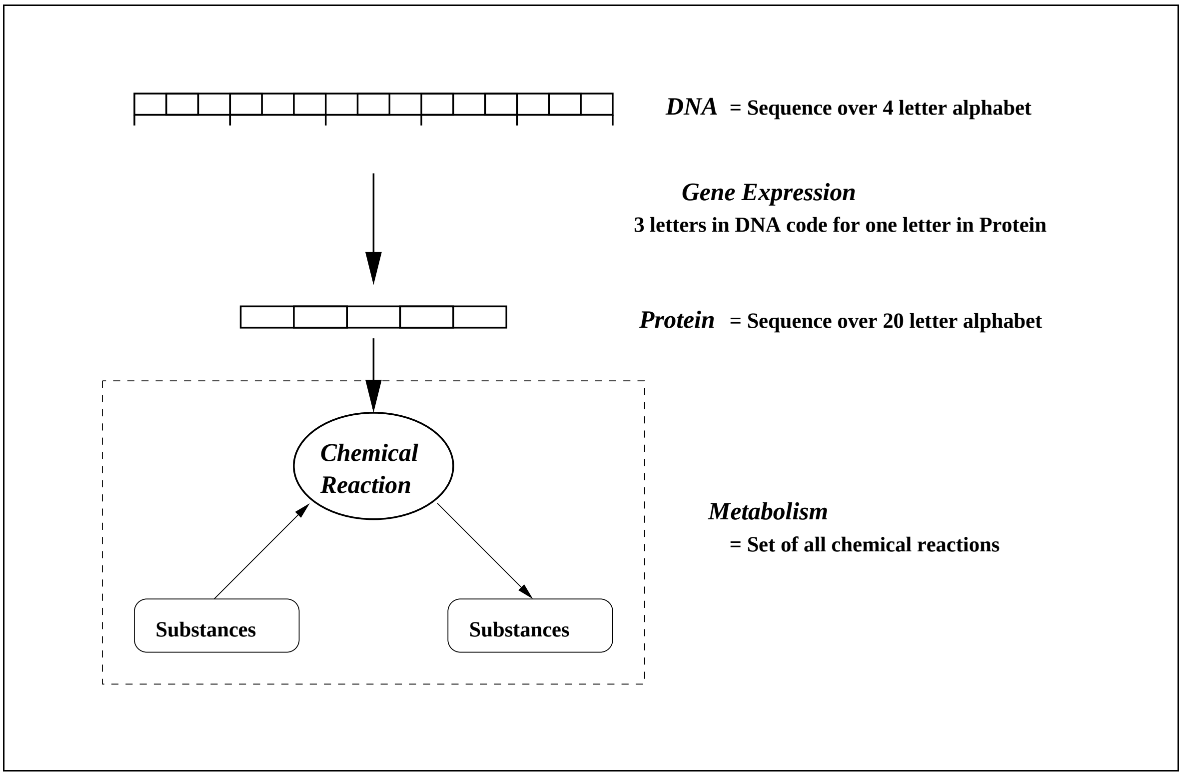


FIGURE 2.1: Overview of the genetic information flow.

2.3 Some Important Macro Molecules

As mentioned above a *molecule* is a set of atoms that are connected by chemical bonds. The term *bio-molecule* usually stands for molecules which occur in living organisms. Primarily, bio-molecules consist of carbon (abbreviated as C) and hydrogen (H). Other atoms such as oxygen (O), nitrogen (N), phosphor (P), or sulfur (S) can also appear.

One way to model molecules is to use (mathematical) graphs. In such graphs, a node represents an atom and an (undirected) edge represents the chemical bond between two atoms. A graphical representation of a water molecule is shown in Figure 2.2 (a). If the nodes are marked with the names of the atoms, the representation is called *chemical formula* (Figure 2.2 (b)).

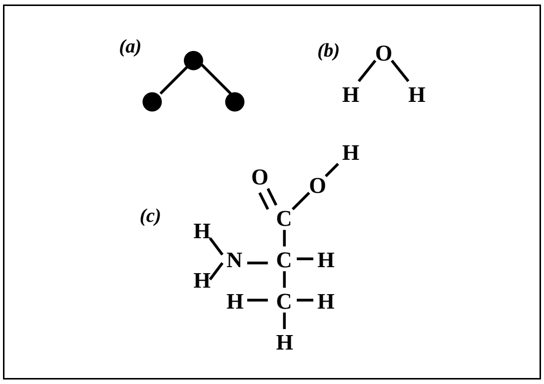


FIGURE 2.2: Schematic representation of molecules: (a) graphical representation of a water molecule; (b) chemical formula of water nodes are marked with the names of the atoms; (c) chemical formula of the amino acid Alanine

A *chemical bond* is based upon the fact that each of the two atoms gives an electron to share it with the other atom. Thus, in a chemical formula each edge represents two electrons which are shared by the connected atoms.

Additionally, connections between two different atoms can also be based on four or even six shared electrons. This is indicated with two (three, respectively) edges between two atoms (see Figure 2.2 (c): the bond between a carbon and an oxygen atom is based on four electrons; it is labelled with two edges).

Finally it can be summerised that a chemical formula is a schematic, graph-theoretic representation of a molecule. Each edge represents two electrons that are shared by the connected atoms. But one should be aware of the fact that this representation is no representation of the 3 dimensional structure of a molecule. This 3 dimensional structure is important because it determines the properties of the molecule.

The atoms can interact with the surrounding molecules as well as with other atoms within the molecule. A consequence of these interactions is that molecules can form bigger complexes by joining each other into a so-called *macro molecule*.

These interactions rely on the 3 dimensional structure of the molecules. It should be pointed out, that the 3 dimensional structure of a molecule is not fully fixed but quite flexible. The flexibility of the 3 dimensional structure of a molecule enables interactions.

TABLE 2.1: Molecular components of a E. coli cell (according to [45])

Component	Water	Proteins	Nucleic Acids	Polysaccharides	Lipids	Others
% of Cell Weight	70 %	15 %	7 %	3 %	2 %	3 %

Molecules can be classified by their atoms and by the way these atoms are arranged (due to the chemical formular). Members of one class usually have similar 3 dimensional structures and therefore form the same macro molecules.

Of course, also the bio-molecules form complex macro molecules as mentioned above. Nearly the whole solid substance of organisms is based upon four different macro molecules called *nucleic acids*, *proteins*, *polysaccharides*, and *lipids*. These macro molecules are (apart water) the most frequent components of cells. See Table 2.1 for a list of all cell components and their proportion of the cell's entire weight.

The rest of section describes briefly the structure and function of these macro molecules.

2.3.1 Nucleic Acids (DNA/RNA)

DNA (= Deoxyribose Nucleic Acid) and RNA (= Ribose Nucleic Acid) are two molecules which are built in a very similar way. Their main function is to store the *genetic information* — the so called *genes* — in cells (cf. Section 2.4.1). Several small viruses (cf. Section 2.4.2) use only RNA for their genes. All other organisms and viruses use DNA.

Both molecules, DNA and RNA, are built out of small molecules called *nucleotides*. There are only five different nucleotides which occur in DNA and RNA. Nucleotides are molecules based on three components: a sugar (ribose or deoxyribose), phosphatic residue, and a base. The base is the variable, characteristic component. Nucleotides are referred to as *adenine*, *thymine*, *cytosine*, *guanine*, and *uracil*, and abbreviated as A, T, C, G, and U — according to their bases. A, C, and G appear in both DNA and RNA. T appears only in DNA and U appears only in RNA. U is the pendant of T in RNA and T the pendant of U in DNA.

DNA and RNA are linear polymers of the five building blocks (A, C, G, T, and U). That means, the sugar and phosphate components of the different nucleotides are linked to a long chain with a regular backbone to form a macro molecule.

The bases are on the sides of this backbone. Although the architectures of DNA and RNA are very closely related to each other, they have small but important differences in both structure and function. One of the building blocks of the backbone are sugar molecules. DNA is based on other sugar molecules than RNA. We will first give more details on the structure of DNA. After that we will turn to the structure of RNA. The question, how these molecules serve as the storage medium for the genes, is discussed in Section 2.5.

DNA The nucleotides that appear in DNA are usually A, C, T, and G. A DNA molecule can thus be seen as a sequence over a four-letter alphabet. This is a common abstraction in bioinformatics.

DNA usually consists of two corresponding chains which are both winded around a common axis to form a *double-helical* structure. The chains are *antiparallel*, i.e. the sequence of the nucleotides of one chain is in reverse order of the other chain. The direction of the sequence is named as follows. One end is called *3' end* (spoken "three prime end"), and the other is called *5' end* (spoken "five prime end"). These names are derived from chemical properties of the sugar component. If a direction is determinated as "in 3' direction", i.e. the end of the chain in this direction is the 3' end. Thus, at each end of the DNA double-helix, one chain provides its 3' end and the other chain its 5' end (see Figure 2.3).

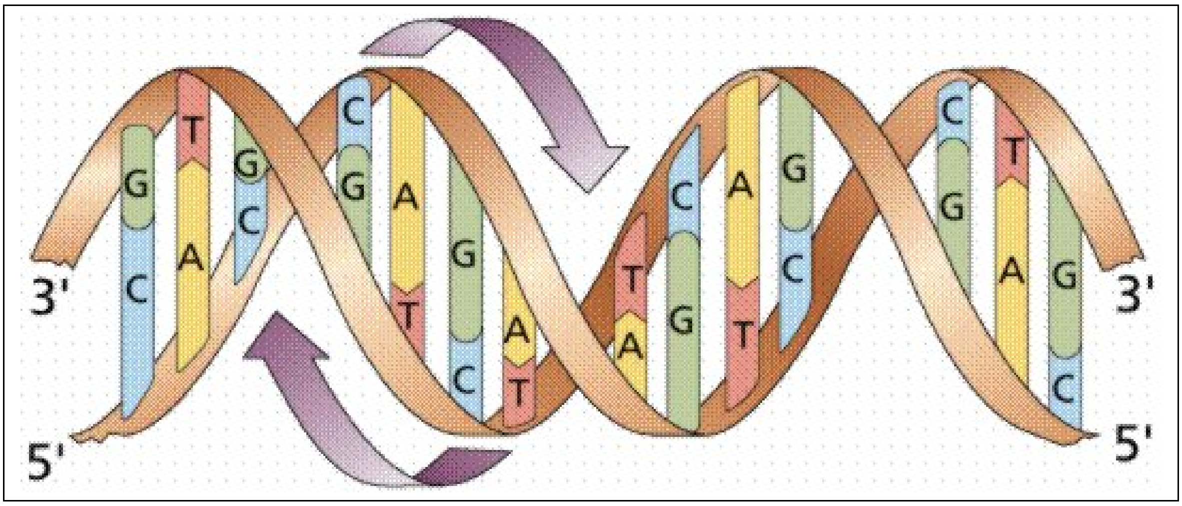


FIGURE 2.3: DNA double helix: the base pairing of the two anti-parallel DNA chains (one chain is in 3' direction, the other in 5' direction) stabilise the double-helical structure of the DNA molecule, as determined by Watson and Crick. (taken from [1])

At each position (position of the bases, that are located on the sides of the backbone) the base of one chain builds a pair with the base of the other chain. A *base-pair* is simply an interaction between the bases standing opposite of each other. These interactions are based on so-called *hydrogen bonds*. Note that hydrogen bonds are no chemical bonds but only tight interactions between two atoms (usually between H and O or between H and N).

The base-pairs between two nucleotide chains stabilise the 3-dimensional structure of the DNA molecule. Base-pairs (bp) are a common unit of measure to determine the length of a complete or only parts of a DNA molecule. A DNA molecule of 100 bp thus consists of two antiparallel sequences, each 100 nucleotides (bases) long. The unit of measure to determine the length of a single nucleotide sequence is called "bases" (abbreviated as "b"), respectively kilo- and megabases (1000b = 1kb, 1000 kb = 1mb).

A very important aspect is that not all base-pairs are possible (allowed). Allowed base-pairs in DNA are the following (replace U for T in case of RNA):

$$\mathrm{T} \sim \mathrm{A} \quad \mathrm{A} \sim \mathrm{T} \qquad \mathrm{G} \sim \mathrm{C} \quad \mathrm{C} \sim \mathrm{G}$$
$$\text{(with } \sim \text{ we denote interaction)}$$

These base-pairing rules, established by Erwin Chargaff, are based on the fact that the pair T $\sim$ A can build two hydrogen bonds. The pair C $\sim$ G on the other hand can build three hydrogen bonds. In living organisms, there are usually no base-pairs beyond the base-pairing rules.

The concept of base-pairing is the most important property of the nucleic acids and plays a fundamental role in the genetic information flow procedures (described in Section 2.5).

RNA In contrast to DNA, the nucleotides that appear usually in RNA are A, C, T, and U. RNA can also be modelled as a string over a four-letter alphabet. Usually, RNA consists of only one chain.

There are different kinds of RNA in living organisms, each with its own structure and function. *Ribosomal* RNAs (rRNA) are structural components of multi-protein complexes called ribosomes (proteins are synthesized at the ribosomes). *Messenger* RNAs (mRNA) are nucleic acids that transport the genetic information from the genes to the ribosomes. *Transfer* RNAs (tRNA) translate the genetic information of the mRNA into the according amino acid sequence. We will investigate the functions of different types of RNA in Section 2.5.

In addition, there are a lot of RNAs with special functions. To list them here would go beyond the scope of this brief introduction. Further information about RNA can be found e.g. in [45].

2.3.2 Proteins

Proteins have many different functions. As so-called enzymes they supervise nearly all cellular reactions. Proteins can also act as anti-bodies, regulatory substances, stabilisers, or carriers of other substances.

Protein molecules are polymers, i.e. consist of thousands or millions of atoms. Special bio-molecules called *amino acids* serve as building blocks of proteins. Amino acids form a long chain with a regular backbone (the so called *polypeptide chain*) and (short) side chains, that are appended to the backbone in regular intervals. A long chain of amino acids is also called *polypeptide*.

It should be mentioned that the backbone as well as the side chains of proteins differ remarkably from those of nucleic acids from the chemical point of view. But the basic architecture principles are similar.

Amino acids consist of a central C-atom (the so called C^{α}). Grouped around the C^{α} there is a *carboxy group* (COOH), an *amino group* (NHH), a H-atom, and a *variable residue*. The universal chemical formula of an amino acid is shown in Figure 2.5 . The polypeptide chain is tied by the N and the carboxy atoms upon loss of one water molecule. The residues differ in length and function. There are only 20 different amino acids that occur in proteins. The amino acids are often classified by the chemical properties of their residues. This is important since the chemical properties of the residue affect the three-dimensional structure (*folding*) and thus the function of the whole molecule. A comprehensive list, including a classification according to the properties of the side chains of the 20 proteins that occur in proteins is provided in Appendix B.2. The *one*- and *three-letter codes* used as abbreviations for the amino acids are also provided.

The three-dimensional structure of a protein is essential for its function. Indeed, the function of a protein depends on how it is folded.

There are four abstraction levels of the three-dimensional structure of a protein (cf. Figure 2.5):

- The *primary structure*: the primary structure of a protein is simply its amino acid sequence. It is often given as a linear string over a 20 letter alphabet representing the 20 amino acids. Usually the sequence is written down from the N- to the C-terminus.

- The *secondary structure*: the secondary structure describes the 3 dimensional conformation of the polypeptide backbone. Probably the best known secondary structures are the α-*helix* and the β-*strand*. Simple combinations of a few secondary structure elements with specific geometric arrangements have been found to occur frequently in protein structures. These units are called either *super-secondary* structures or *motifs*.

- The *tertiary structure*: the tertiary structure describes the three-dimensional structure of a polypeptide chain and its residues, i.e. their relative positions in a 3 dimensional space. Folding

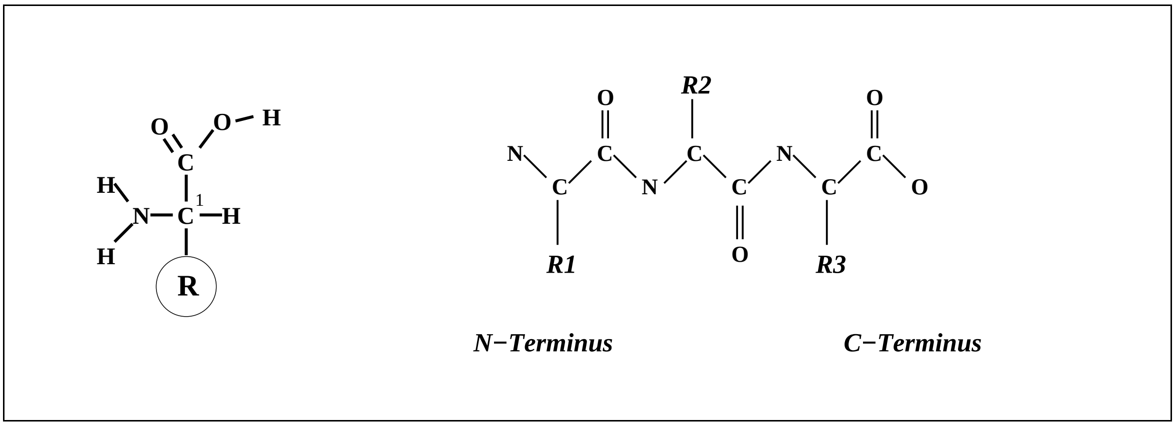


FIGURE 2.4: Left: chemical formula of an amino acid prototype. The 20 different amino acids, that appear in living organisms, differ only in their residue (R). The participating atoms tying the polypeptide chain are the N and the C^1 (C^α) atoms. Right: a polypeptide chain of three amino acids with variable residues R1, R2, and R3. The residues appear on the sides of the polypeptide chain. One end is called the N-terminus, the other end is called the C-terminus.

units within one polypeptide are called *domains*.

- The *quarternary structure*: several proteins are built upon more than one independent polypeptide chain that interact with each other to form a bigger protein complex. The single polypeptide chains are called *subunits* in that context. The quarternary structure describes the arrangement of these subunits (E.g. hemoglobin, the oxygen carrier in the human blood, is based on four subunits that interact with each other to form the entire protein complex).

The primary, secondary, tertiary, and quarternary structures form a hierarchy of abstractions describing the structure of proteins from the simpler to the more precise (complex).

Sometimes these terms are also used in the context of other macro molecules, especially nucleic acids. The primary structure of a nucleic acids molecule determines the sequence of its nucleotides. The secondary structure of DNA is a double helix (cf. above). RNA can form different secondary and tertiary structures.

Several forces, that act inter-molecular (between two distinct molecules) and intra-molecular (within a same molecule), stabilise the three-dimensional folding of proteins. The most important forces are called hydrogen-bonds (these forces stabilise also the double-helical structure of DNA), di-sulfide bridges, polar interactions, and hydrophobic forces.

In general, a protein folds into the correct conformation spontaneously after it's aggregation. The folding pathway (i.e. the blueprint of the reactions and procedures of protein folding) has been solved for several simple polypeptides. However, for lager proteins and especially for those with several folding domains, the folding gets very complex and several pathways lead to the identical final structure. Specialised proteins called *chaperones* usually assist the folding of proteins. This makes the folding procedure even more complex.

To predict the three-dimensional structure of a protein from its amino acid sequence (i.e. from its primary structure) is thus an unsolved problem. This problem is known as the *protein folding problem* and is one of the major problems in structural molecular biology. Several computer-based approaches have been proposed to help solving this problem, but a universal computer program to predict the

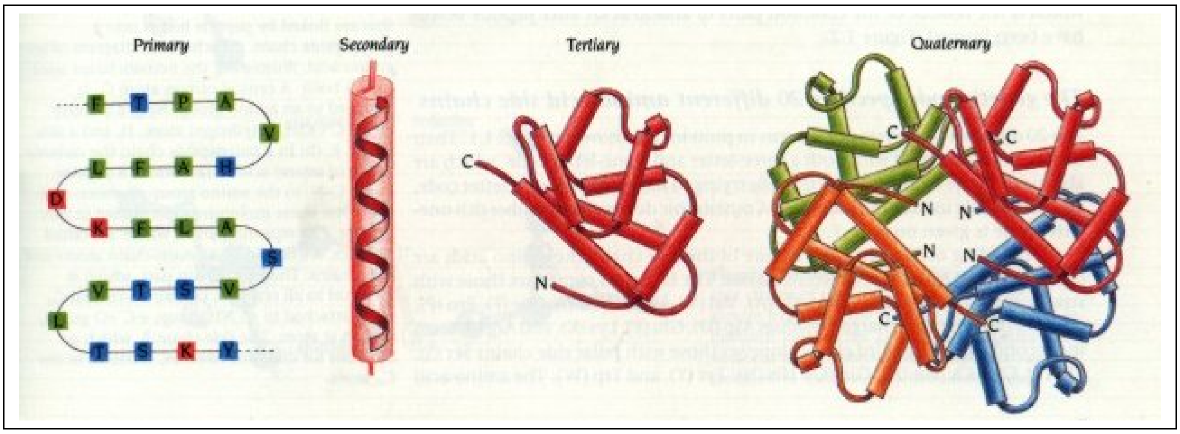


FIGURE 2.5: The four abstraction levels of protein structure: The primary structure determines the amino acid sequence; the secondary structure determines the folding of the backbone (polypeptide chain); the tertiary structure determines the folding of the polypeptide chain and of the residues; the quarternary structure describes the arrangement of several subunits (taken from [9])

protein structure from its amino acid sequence does not seem to be in sight so far.

2.3.3 Polysaccharides

Polysaccharides have several functions. The primary function of polysaccharides is the storage of energy. They can easily and quickly be processed to their building blocks. A result of this procedure, that transforms e.g. glycogen into glucose and further the glucose into small molecules, is biological energy.

Another important aspect of these macro molecules is, that they are used as structuring elements in cell walls (e.g. cellulose).

The building blocks of polysaccharides are the so called *monosaccharides* (or *sugars*). Probably the best known of these bio-molecules are glucose and fructose. Monosaccharides, that occur in macro molecular complexes, are usually chains of four to six carbon atoms. Usually each C atom has a bound oxygen atom. Monosaccharides with five (*pentoses*) and six (*hexoses*) C atoms form rings. E.g. the sugar part of nucleotides is the so called ribose (in RNA), or deoxyribose (in DNA). Both molecules are pentoses.

Monosaccharides can be connected to long chains, the polysaccharides. An example is cellulose which is built up by many glucose units bound to each other.

Saccharides can also interact with other macro molecules. For example, they can form complexes with proteins, the so called glyco-proteins.

2.3.4 Lipids

The most important property of (and criterium for being classified as) a lipid is that it is not soluble in water. The main functions of lipids are the storage of energy and the construction of isolating components like membranes. Membranes usually consist mainly of lipids (and other macro molecules) forming chemical and electric seperated areas — organelles and cells.

2.3.5 Summary

The architecture of all macro molecules is related: Macro molecules are built upon small molecules, the bio-molecules. We have briefly summerised the structure of the four most important macro-molecules and their building blocks.

Nucleic acids (DNA/RNA) are made of (usually) four different nucleotides. An abstract model of a DNA or RNA molecule is a string over a four letter alphabet. The concept of base-pairing is the most fundamental property of nucleotides.

Proteins usually are made of twenty different amino acids. The amino acid chain (polypeptide back-bone) folds into a complex three-dimensional structure. There are four different abstraction levels describing the three-dimensional conformation of a protein in more or less detail. It is the three-dimensional structure that determines the function of a protein.

Intra- and intermolecular forces stabilise the conformation of proteins and nucleic acids. The most important force is called hydrogen bond.

Polysaccharides are primarily used by all organisms to store energy. In several successive reactions these macro molecules are processed and fractionalised. These reactions are supervised by enzymes (proteins) and form the so-called metabolic pathways (cf. Chapter 5).

Lipids are molecules with different structures that have one basic property in common: They are all not soluble in water. Lipids usually have a polar end (which can interact with water molecules and would be soluble) and a long hydrophobic end (which is responsible for the non soluble character of the macro molecule).

2.4 Supra-Molecular Complexes, Cells, and Cell Parasits

The macro molecules mentioned in Section 2.3 interact with each other to form many *supra-molecular complexes*.

Supra-molecular complexes are essential components of all cells.

2.4.1 Cells

Cells are the central organisation units of all living organisms. They share several properties but cells of different species or different cells of the same organism can differ remarkably in structure and function.

Cells are complex systems. They separate themselves from their surrounding by a membrane (called *plasma-membrane*), withdraw their surrounding environment raw materials, are built up from this raw materials and regulate their own growth and reproduction.

Some organisms consist of only one cell, e.g. bacteria or the bakers yeast *S. cerevisiae*. Others are formed by many cells that often fulfill different functions and specialised structures, which interact in a complex way. The human organism for example consists of more than 10^{14} cells.

The lumen of the cell is a kind of wet gel called *cytoplasm*. Several cells have different substruc-tures, which are separated from the cytoplasm by membranes. These compartiments are called (cell)*organelles*.

Living organisms are divided into two classes according to the properties of their cells, *prokaryotes* (bacteria) and *eukaryotes* (all others). Whereas bacteria (prokaryotes) normally do not have any organelles, all other organisms (eukaryotes) contain a specialised organelle containing the DNA, called the nucleus. Besides the nucleus, eukaryotes have other organelles, e.g. the mitochondrion which produces energy for the cell out of chemical substances.

The complete amount of genetic information in a cell is called the genome. It consists of genes which are linear sequences of nucleotides in the DNA.

The genome determines what proteins can be produced by the cell, which chemical reactions the cell can provide, what macro-molecules the cell can produce. Summarising we can say that the genome is the storing-medium that determines the functional abilities of the cell.

Procedures called transcription and translation are involved in the production of a protein out of a gene. The sequence of nucleotides determines the sequence of the amino acids of the corresponding protein.

For reproduction the cell divides itself into two similar successor-cells. Every successor-cell gets its own copy of the genome of the original cell. This procedure is called replication.

These procedures are described (in more detail) in Section 2.5.

2.4.2 Cell Parasites: Viruses

There is another organisation unit with functions and structure different to cells. Particles called viruses have also their own genome. But in contrast to cells they have not the ability to reproduce their genetical information on their own. They need suitable *host cells* for reproduction. Hence, viruses are normally not counted as living organisms.

Viruses are packages of nucleic acids which are covered by a protein envelope. The size of viruses is much smaller than that of cells (mostly about 100 times smaller). The complexity of these particles ranges from viruses with only four genes up to viruses with about 250 genes. The viral genome can be stored in DNA or RNA. Viruses do not have the ability to replicate or translate genes on their own. They use the equipments of cells to reproduce its genome or produce proteins. This is often done by inserting the viral genes into the genome of the host cell. As a consequence, viruses are also called cell parasites.

To reproduce themselves, viruses have to get into a host cell. This procedure is called (viral-)*infection*. Viruses are more or less specific for a special host cell. Bacteriophages e.g. are viruses that infect bacteria.

A viral infection of a cell consists several steps:

1. The virus has to insert its own genome (mostly DNA) into the cytoplasm of the cell.

2. The viral genome is replicated, transcripted, and translated (cf. Section 2.5 for details) by the cell. This means, that the components to build new virus particles are produced. Several new virus particles are created.

3. Finally these newly produced viruses leave the cell. The host cell is often killed during this step. The new viruses are ready to infect other cells.

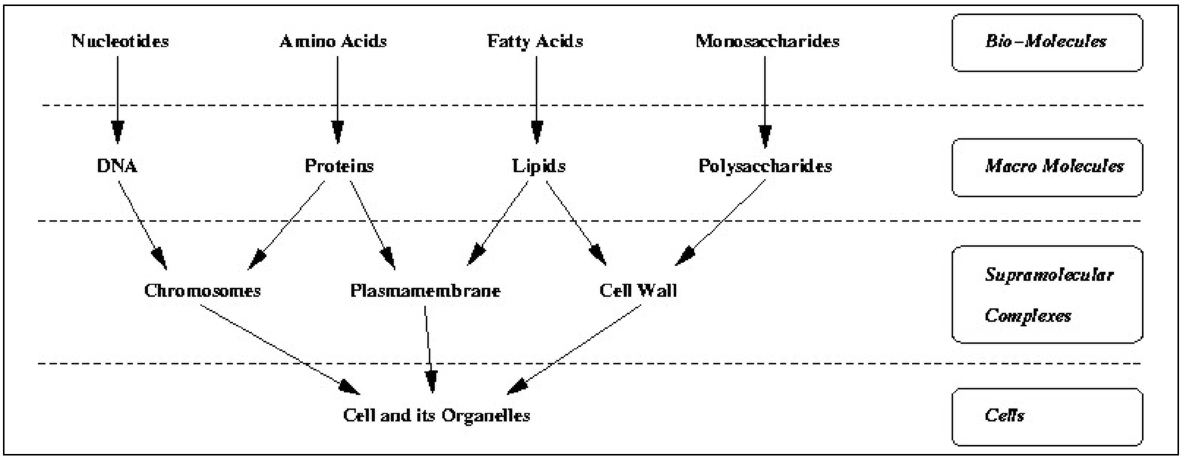


FIGURE 2.6: Hierarchies of organisation in living organisms

Some viruses are dangerous for humans. E.g. influenza or AIDS are caused by a viral infection. Nevertheless, not all viruses cause diseases. Viruses can be used as DNA carrier particles in genetic engineering (cf. Section 2.6).

2.4.3 Summary

Different levels of the organisation hierarchy of living organisms were described (see Figure 2.6).

Cells are the central organisation units of living organisms. They perform many chemical reactions each supervised by enzymes. The enzymes are determined in the genes and are produced by the cell itself. We will investigate this production in the following section.

Viruses use the equipment of the cells to reproduce their genes. This is called infection. Viruses can cause special diseases. They are sometimes used as carriers for genes in the field of genetic engineering (cf. 2.6).

2.5 The Genetic Information Flow (More Detailed)

In this section the following topics will be discussed: How can organisms produce nearly identical copies of themselves and many identical copies of big and complex macro molecules?

At the beginning of the 19th century, researchers guessed that there must be a matrix or a template, and they were right. DNA usually serves as this template. It is the matrix for the identical reproduction of itself (*replication*). And it is usually the matrix for producing many identical copies of complex proteins. The procedures to synthesise a protein out of a gene are called *transcription* and *translation* and are based on the principles of the base-pairing rules.

2.5.1 Replication

With the determination of the DNA structure by James Watson and Francis Crick in 1953, it became clear how DNA is used as a template to reproduce the genetic information.

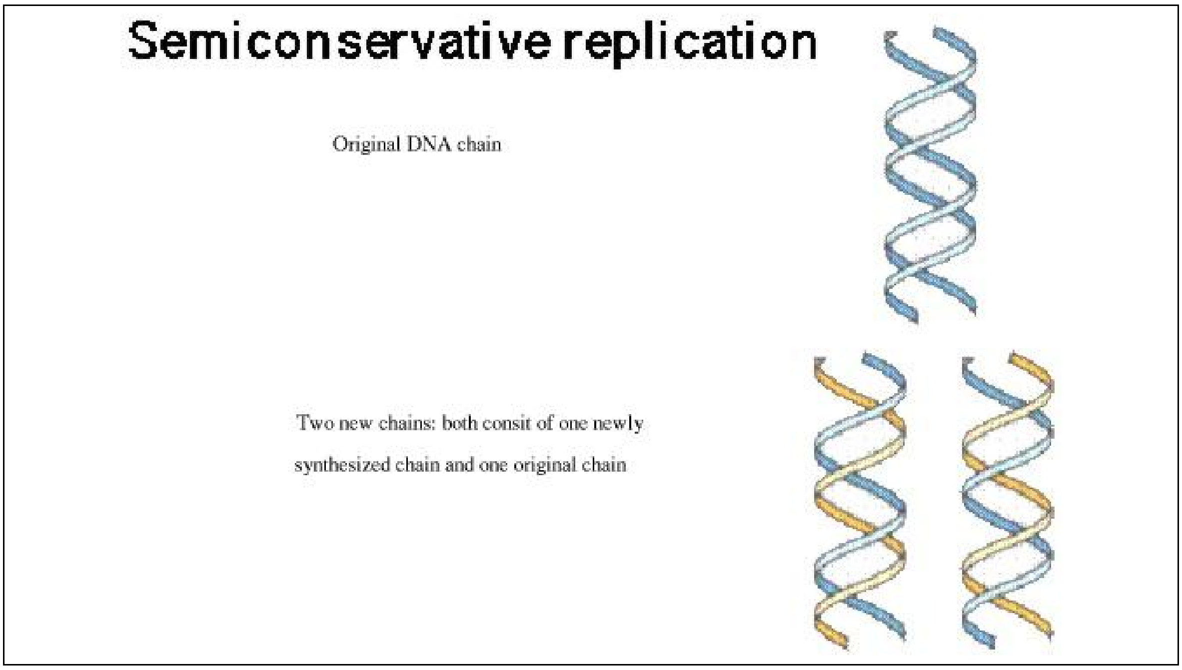


FIGURE 2.7: Semiconservative DNA-replication: each original chain is a template for a new chain. The base-pairing rules ensure, that the new chains are identical copies of the original one.

The fundamental assumption is, that each DNA chain is an identical copy of the other one.

Based on the strict rules of base-pairing one chain can be the template for a new chain. This new chain will be anti-parallel to the original chain.

The original chain is read. At each position of the new chain, the base corresponding (according to the base-pairing rules) to that of the template is appended. See Figure 2.7 .

The replication of DNA follows some rules: First, the replication is *semi-conservative*. That means, that two new DNA strands are produced from one double strand. Each new chain consists of one original and one new chain. Second, the replication starts at a specific starting point in a DNA molecule (the so called *origin*). Usually it moves on in both directions then. Third the replication usually goes through several phases, such as initiation at the origin, elongation of new chains, and termination. Many different proteins take part on the replication procedure. Nevertheless, the replication has to be – and indeed *is* – very exact.

In fact, eukaryotic replication is much more complex than the prokaryotic replication, but follows the same rules.

Cell reproduction is a complex procedure. One step is the replication of the genetic information. Usually a cell divides itself into two succeeding cells. Each successor-cell gets its own (identical) copy of the original DNA molecule.

In the context of DNA replication, the DNA repair mechanisms should be mentioned. In fact, there are many different impacts that can cause DNA damages. Therefore, a cell has several sophisticated tools and mechanisms to repair damaged DNA.

A probably well-known damage is *mutation*. Nucleotides within the base-sequence can be deleted, substituted, and/or inserted. If any of these changes in the sequence is inherited to the successor cells, this is called mutation.

2.5.2 Transcription

Transcription is the first step of the information flow from gene to protein.

The information stored in DNA is transformed into a RNA sequence. Transcription occurs in the 5' to 3' direction, where the DNA nucleotides A, C, G, T are transcribed into the corresponding RNA nucleotides U, G, C, A. Many different enzymes are participated in this procedure. The RNA produced by transcription is called messenger RNA (mRNA).

Generally, there are three phases. First, the initiation: the transcription starts at a special DNA segment called *promotor*. This promotor usually has several AT subsegments (e.g. TATAA…), and is thus often called *TATA-box*. Second, the elongation of the new chain: only one DNA strand is read (is used as a matrix). This chain is called the coding strand. The newly produced RNA chain is thus identical with the second (non-coding) DNA strand. Last, the termination of replication procedure is initiated by an other signal sequence (the *terminator*).

In prokaryotes, the reading frame of the transcription of a gene is continuous, i.e. not interrupted. Eukaryotes on the other hand, are different. The reading frame of the transcription is frequently disrupted by so called *introns* (or intervening sequences). The remaining segments between the introns are called *exons*. The introns in the RNA transcript are excised and removed after transcription. This procedure is called *splicing* (see Figure 2.8). Note that it is an important algorithmic problem in bioinformatics to determine intron/exon splice sites.

After removal of introns, it may occur that nucleotides are inserted or deleted (each pointwise). This is called *RNA editing*, and probably directed by *guide* RNA (gRNA). This editing, occurring in most living organisms, can be very intensive.

The protein encoded by the mRNA may be completely unrelated to the protein that would have been encoded by the RNA before editing. The reason for this is, that the *genetic code* (cf. Appendix B) is a triplet, non-overlapping code. An insertion or deletion of one nucleotide shifts the original reading frame. After splicing and editing the mRNA is often called mature.

2.5.3 Translation

After the description of the transcription process, the following question will be addressed: How is the transition of information from the language of nucleotides to the language of amino acids mediated?

Organisms use a helper molecule called transfer RNA (tRNA). This tRNA is a small (appr. 70 bp) macro molecule. It has a cloverleaf secondary structure (backbone of the chain) and an L-shaped tertiary structure (overall three-dimensional structure). At the extremity of the middle lobe of the cloverleaf lies an *anticodon*. An anticodon consists of three nucleotides, whose reverse complement (according to the base-pairing rules) is a triplet codon for an amino acid (as given in Appendix B). On the opposite lobe of the cloverleaf, there is the corresponding amino acid attached to the tRNA molecule, i.e. the amino acid, whose codon is the reverse complement of the anticodon of the tRNA.

The translation is performed by the ribosomes. Ribosomes are supramolecular complexes based on ribosomal RNA (rRNA) and proteins. The ribosomes provide a surface for mature mRNAs and tRNAs to assemble amino acids together to linear chains.

The chains are extended as described in the following. The mRNA walks along the ribosomes with a three-nucleotide step size. tRNA providing a fitting anticodon,is hybridised onto the mRNA. The amino acids bound to the RNA are then transfered onto the growing (C-)terminus of the nascent polypeptide chain.

This process is sequentially repeated until the translation process reaches a stop codon, which terminates the polypeptide synthesis. Appendix B provides a table summarising the genetic code (Appendix B depicts which codons of mRNA determine which amino acids).

2.5.4 Summary

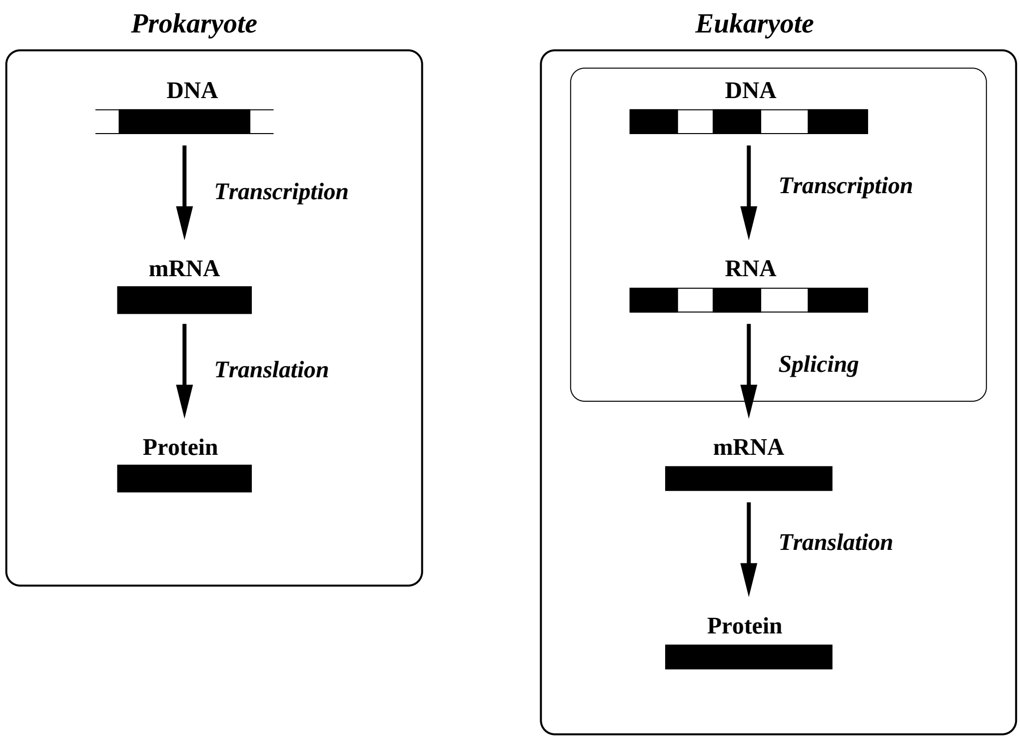


FIGURE 2.8: The different models of transcription and translation in prokaryotes and eukaryotes (taken from [18])

The genetic information of cells is stored in genes as a linear sequence of nucleotides. A procedure called replication enables cells to produce an identical copy of its own genome.

The genetic information is expressed through procedures called transcription and translation. The linear sequence of nucleotides is translated into a linear sequence of amino acids. Three nucleotides in DNA code for one amino acid. The amino acid chain determines a protein.

Many proteins supervise specific metabolic reactions. These proteins are called *enzymes*. An enzyme usually transforms a substance (substrate) into another substance (product).

It should be mentioned, that some viruses can perform a procedure called reverse transcription. This procedure transforms RNA into DNA. Some viruses use RNA to store their genome (and not DNA). But the cellular mechanisms of translation and transcription are specific to DNA, i.e. that they can only express genes stored as DNA-sequence. As viruses have to use the mechanisms of their host cells, they have to transform their RNA-genes into DNA-genes to express their own genome.

2.6 A Selection of Biology and Bioinformatics Applications Related to Molecular Biology Databases

2.6.1 Model Organisms

As mentioned above, the complete ”Entity-Relationship-Model of life” is far out of reach. But the molecular interactions follow common laws, although each organism has its own variants.

As a consequence researchers can make conclusions from information about one special organism regarding other related organisms. There are several well-investigated organisms. These are called model organisms. Data and information of another organism related to a model organism is compared and may be derived from knowledge about the model organism.

An interesting and important example for this is the following situation: A researcher may have detected a gene in the human genome. The gene will be translated into a protein, but nobody knows its conformation and function. The researcher will visit some molecular biology databases and search for the gene and the protein in other model organisms. A gene with a high value of similarity may be found in the *S. cerevisiae* genome, which is completely sequenced, from which the function of the encoded protein is known. The researcher will conclude with a rather high probability that the human protein, which was detected, has the same function. In addition, it can easily be tested whether the human protein can functionally replace its fungal homolog as the protein from the other organism that has similar properties is called (one speakes of “*reverse genetics*”). Computer scientists might compare this kind of information transfer with a knowledge-based approach to a specific problem.

Two of the probably most important model organisms are the bacterium *Escheria coli* (procaryotic model organism) and the baker’s yeast *Saccharomyces cerevisiae* (eucaryotic model organism). The *Escheria coli* organism is one of the best discovered procaryotic organism. Yeast is one of few eucaryotic organisms that consists of only one cell and is thus less complex as multi-cellular organisms. The genome of both organisms is fully sequenced (i.e. determined).

2.6.2 Essential Methods for Generating Biological Data

DNA-Cloning

DNA-cloning was introduced in the early seventies. It revolutioned molecular biology and established genetic engineering as a central method in molecular biology. It allows to generate any desired alteration of the genetic information of an organism and to selectively grow this alteration in pure (homogeneous) cultures (*clones*).

DNA-cloning usually consists of five steps [11]. In the following we assume that the DNA segment which should be cloned is a gene called *X*:

1. A DNA-fragment (containing the gene *X*) has to be inserted into a ring-shaped DNA-molecule

(the so called *plasmid*).

2. The plasmid serves as a gene-carrier. It transports the gene *X* into a host cell.
3. The plasmid is reproduced in the host cell or can be integrated into the host cell's genome. Thus many identical copies of the gene *X* are produced.
4. If the host cell reproduces itself, copies of the gene *X* are inherited to the successor cells. In these successor cells the gene is reproduced again.
5. Many cell reproductions produce a clone. Each cell of the clone contains at least one copy of the gene *X*.

DNA-cloning requires several tools and methods. One should be mentioned here: Step one of the cloning procedure requires methods to cut and tie DNA segments. That can be done by special enzymes, called *restriction-endonucleases*, that cut DNA at specific positions. To ligate two DNA-fragments a enzyme called DNA-ligase is used.

Genetic databases focussing on information on a specific organism often provide important information about these tools and methods for DNA-cloning. Sometimes, genetic maps are offered. These maps are graphical representations of the genome providing the locations of each gene and the positions where endonucleases can cut the DNA.

DNA-Sequencing

One of the most important and productive applications in the field of molecular biology are sequencing methods. These methods aim to explore the primary structure of a DNA-molecule by determination of nucleotide sequence. Currently there are many efforts to determine the whole genome of many different organisms. The most important and well-known may be the *Human Genome Project* that aims to explore the sequence of the human genome. There are two different methods to determine a DNA sequence – the Sanger-Coulson-Method and the Maxam-Gilbert-Method. Both methods can determine the sequence of about 400 nucleotides per reaction. Most of the genes and especially complete genomes of even simple organisms are much longer, e.g. humans have 3×10^9 bp/genome. For sequencing a gene or genome researchers have to perform one of the sequencing methods several times. The sequence of small segments with overlapping ends have to be determined. The whole sequence has to be recombined from these segments manually or by using computational tools. The determination of a complete genome of an organism is called "sequencing". The complete genome is "sequenced".

Sequencing can be performed rather easily. As a consequence research groups all over the world produce large amounts of sequence data. The sequence data are stored in sequence databases (see Chapter 3). An entry of these databases often consists only of the sequence of a special gene or of a sequence whose biological functions are not yet known. These entries include:

- *Expressed Sequence Tags* (ESTs): ESTs are short (200-500bp) DNA sequences generated from randomly selected cDNA clones. The purpose of EST-sequencing is to scan rapidly for all the protein coding genes and to provide a tag for each gene on the genome.
- *High Throughput Genomic Data* (HTG): HTGs are segments of sequences (typically more than 2kb long) which are unfinished, that means, in general, contain one or more gaps and other errors or are unordered.

```
S1: A T C G T C A T G G G C T
          | | |   | |   |
S2:     A G T C T T G A G
```

FIGURE 2.9: Alignment of sequences S1 and S2: best matching of the nucleotides of S1 and S2

2.6.3 Typical Bioinformatics Applications Related to Molecular Biology Databases

The amount of software tools for analysing molecular biology data is overwhelming. But there are some standard tools, that are provided by several databases. These tools support frequently appearing analysis requests.

The probably two most important methods are described in the following.

Sequence Analysis

Nowadays sequence analysis is perhaps *the* most important bioinformatics research field. A large amount of sequence data is available and is increasing with an exponential speed.

Sequence analysis methods have different goals, amoung others:

- **Pattern Detection:** Newly produced DNA sequences are just strings over a four letter alphabet. The biological function of the sequence is usually not known. With the support of sequence analysis tools these sequences have to be analysed for detecting common patterns (e.g. a start codon indicating the start of a gene, or a sequence pattern indicating a regulatory region).
- **Sequence Similarity:** Newly produced data are also often compared with existing sequences. Sequence similarities indicate similar biological functions.
- **Gene Translation:** If a sequence is identified as a gene, it is translated according to the genetic code (cf. Appendix B) into a amino acid sequence.

Especially the sequence similarity search is a common method. In this context the term *sequence alignments* is often used.

The sequences S1 and S2 are matched in Figure 2.9, according to their nucleotides. The best alignment is the alignment optimizing the number of matchings. The best alignment of sequences S1 and S2 is shown here.

Common sequence analysis methods used by molecular biology databases are the BLAST [4] tools (BLAST is a family of database retrieval tools based on alignment algorithms; BLAST can e.g. find the best alignment of a given sequence and all sequences in a sequence database) or FASTA [56] (a sequence alignment tool with similar functionality like BLAST). Both tools require input as text files in a special format.

Protein Folding and 3D-Structure Analysis

As mentioned above, the precise prediction of the 3-dimensional structure of a protein from its amino acid sequence is an important problem for biologists.

Several approaches have been developed to solve this problem and create an algorithm that predicts secondary structures from primary structures.

These approaches usually try to create an oversimplified model. E.g. common oversimplifications are the exclusion of exceptions or the assumption that the protein structure is not flexible but fixed. If the model is accurate, the most possible folding can be calculated. But the calculated conformation has to be checked manually in the laboratory by biologists.

Nevertheless, this procedure of calculating and checking is usually much quicker than solving the 3-dimensional structure manually in the laboratory without calculating the most possible conformation before.

The most frequent used approaches of structure prediction and analysis are the knowledge-based approaches. It is a fact that proteins with a similar amino acid sequence usually have similar functions and therefore also similar structures. Newly discovered or translated proteins are compared with existing proteins (of which the structure is known) by using sequence alignment methods.

3-dimensional structure analysis tools that are with molecular biology databases have different functionalities. There are tools for the visualisation of protein structures (e.g. RasMol [60]). A tool called VAST is a structure – structure alignment method (i.e. aligns two proteins on the sequence and the structure level). A long (but probably not comprehensive) list of 3-dimensional structure analysis tools can be obtained from the homepage of the Protein Data Bank PDB at: `http://www.rcsb.org/pdb/software-list.html`

Note: The primary structure of nucleic acids (i.e. the sequence of nucleotides) is comparable to a machine-readable language. The “translation” of this blueprint into a folded protein is extremly quick. If this “translation” would be fully understood, this would probably lead to a new computer model in terms of efficiency. This computer model would probably not be comparable with the traditional models (such as turing-machine), but instead be much more efficient.

Chapter 3

Molecular Biology Databases

In the past few years the systematic sequencing efforts on all kind of organisms have grown rapidly. A tremendous flood of biological data is now produced. As biology is a knowledge-based discipline and the amount of existing data is overwhelming, applications and systems have to be developed for both storing and warehousing all these data. It is no wonder that there are many different molecular biology databases which were established in the past years. These databases are often built and maintained by biology researchers who do not have profound expertise in the field of computer science and especially database design and applications. On the other hand these molecular biology databases have been neglected by a great part of the computer science community for a long time.

This part of the work is dedicated to computer scientists to get insight into molecular biology databases. We try to investigate the properties of these databases, focussing on the data modelling, the data storage, and the data retrieval. A classification of molecular biology databases according to these issues is provided. In Appendix A we present a directory of 120 molecular biology databases. Each listed database is investigated according to the classification criteria mentioned above.

Chapter 3.1 gives a brief introduction to molecular databases followed by a description of the criteria used to classify the databases in Chapter 3.2. Chapter 3.3 describes the four categories of the data modelling classification in more detail. The question for which applications biology researchers use molecular biology databases is discussed briefly in Chapter 3.4. A table that summerises the classification of the molecular biology databases investigated in Appendix A is shown in Chapter 3.5.

3.1 Introduction

In the past few years the systematic sequencing efforts on all kind of organisms have grown rapidly. An overwhelming flood of sequence data is produced. This data will become more and more important as sequence-based methods get constantly fashionable in the field of molecular biology. Efficient methods and resources have to be developed for both warehousing and accessing this tremendous amount of data. As a consequence many databases for this data were established in the past years (e.g. in April 2001 the current version of DBcat – a catalog of molecular biology databases, cf. Appendix A.19 – provides 511 different resources). These databases are mostly accessible via the

WWW.

Unfortunately the field of molecular biology is very complex. Computer scientists usually have no profound expertise in this field. Thus the databases are mostly developed by biologists who themselves have only poor knowledge in the field of databases and data warehousing. It is an important and difficult task to bring the researchers of both molecular biology and computer science together.

This part of the work tries to address this task. We investigate the organization and structure of molecular biology databases. Our focus is on data modelling, data storage, and data retrieval. One should remember that this research field is very dynamic and parts of this work might be out of date soon. But we hope to provide a starting point for beginners in the field of modelling biological data.

We classified the databases in three classes. The first classification is by contents. The second classification is according to the implementation, especially according to the underlying database management system and data model (Chapter 3.2.2 Implementation Aspects: Data Modelling, Data Storage and Data Acquisition). The third classification is according to the data representation. We denote a particular interest on the data retrieval and on query interfaces (Chapter 3.2.3 Implementation Aspects: Data Retrieval/Query Answering).

Appendix A presents an overview over 120 molecular biology databases. The selection of the investigated databases is close to the Database Issue 2000 of the biological journal *Nucleic Acids Research* (NAR, 2000, Vol.28, No. 1). We explore the databases according to the classification criteria described later in this chapter. High value is set on providing literature references for further and detailed investigations.

Chapter 3.5 summerizes the classification of the investigated databases in a table.

3.2 Classifying Molecular Biology Databases

We selected three major classifications of molecular biology databases.

The first classification is according to the contents of the databases. In this context we will take a look on the interconnections between molecular biology databases (i.e. how are entries of one database linked to entries in other databases.)

The second classification focuses on the implementation. We investigate the underlying data model and data management system. The way and frequency of acquiring new data items is also analysed.

The third classification explores the possibilities of data retrieval. We focus our investigation on the query interfaces of the databases.

This selection seems to be reasonable. Classification by contents is commonly used and provided by biologists. The later two classifications analyse the modelling and representation of biological data and are more intended to computer scientists.

3.2.1 Biological Data

In general, knowledge bases in molecular biology must capture three types of data (see also [57]):

1. Data on biological entities such as nucleic acids, proteins, etc. and on relationships between these entities (descriptive knowledge).

2. Data on the dynamic behavior of biological processes that consit of these relationships (behavioral knowledge).

3. Data on the methods which can be used to identify the entities and their relationships (dynamic knowledge).

As a consequence biological data are rather heterogeneous. On one hand there is a huge amount of data on different biological entities, such as nucleic acids, proteins, genes, etc. On the other hand, there is a tremendous amount of data on many different organisms. Compared to this ever increasing flow of data, only few data are curently available on interactions between biological entities and the dynamical behavior of systems of interactions (such as genetic regulatory systems and metabolic pathways (see also Chapter 5). Depending on their scopes and aims, molecular biology databases usually focus on one of these kinds of data. Databases providing data on biological entities and/or their interactions usually try to gather data from all kind of organisms. This heterogeneity is most likely the main reason for the large and still increasing number of molecular biology databases.

The classification of molecular biology databases by contents is interesting and helpful especially for biologists. Most computer scientists do not have a clue about the contents and organization of the databases. We try to address this problem in the following.

It should be mentioned that every classification of molecular biology databases by contents is questionable from the biological point of view. Usually each biological expert has its own classification according to its main focuses. Our classification by contents is chosen to hopefully enable computer scientists to get a notion about the contents of molecular biology databases.

Biological data often consists of the so called core data. These data are mostly sequences (nucleotides or amino acids). Usually databases provide annotations to these core data. In general these annotations are simple texts, telling biological relevant information about the sequence data. Often, new data, which also have to be stored, are produced out of these core data by dedicated analysis tools provided by bioinformatics research.

Molecular biology databases can be classified by contents in nine main groups. The groups are not disjunct, and databases may be added to more than one group. Each group is described now:

Bibliographic Databases. Bibliographic databases abstract the scientific literature in machine readable form. It should be emphasized that none of these services has a complete coverage of the field of molecular biology.

Databases that Classify Species (Taxonomy Databases). Databases that classify species are often called Taxonomy Databases. They intensively depend on the classifications which are done by a taxonomist. As the classifications are questioned by other taxonomists these databases are fairly controversial.

Nucleotide Sequence Databases. Nucleotide sequence databases focus on biological entities such as genes, nucleic acids, etc.

In general, these databases intend to make freely available all nucleotide sequence data (DNA, RNA) from the research community. Nucleotide sequences (DNA/RNA) are mostly annotated with text (e.g. texts telling in which organisms the sequence appears or which physiological functions the sequence have) and bibliography reference. Some specialized databases deal with particular classes of sequence, e.g. DNA, RNA, transcription factors, HIV sequences, promoters, etc. or try to address the redundancy problem of similar sequences (sequence-cluster databases). The user of such databases should consider that the resources often contain many errors. Although the nucleotide sequence data is often checked by the database staff, the correctness and quality of the data is in the responsibility of the submitter. Unfortunately many data items are mislabeled, contaminated, incompletely or faulty annotated, contain sequencing errors or are simply redundant with other existing entries.

Genomic Databases. Genomic databases in general provide genetic data about a special organism of interest (often model organisms, cf. Section 2.6.1) but vary greatly in content. Information stored in genomic databases includes e.g.: information on genotypes, gene products, gene names, gene properties, specific mutations as well as genomic maps and reference information for strains. The long list of model organisms range from the bacterium *E. coli* and the bakers yeast (*Saccharomyces cerevisiae* to several plants (e.g. maize, cotton, rice, etc.) and animals (e.g. fruit fly, mouse, sheep, etc.).

Proteomic Databases. Proteomic databases are in general a mixture of nucleotide sequence databases, protein sequence databases, and others. They focus on information about expressed proteins, providing usually the amino acid sequences, the nucleotide sequences of the according genes, and biological information about the proteins. This includes maps resulting from 2D electrophoresis (a special biological analysis method) of the according protein. Proteomic databases provide data on biological entities.

Metabolic Pathway Databases. Databases on metabolic pathways store information on the metabolism of one organism or several different organisms. The enzymes participating on the reactions of the pathways are often linked with sequence databases or annotated with biological information. The contents in these databases can be data on biological entities as well as on different species.

Protein Sequence Databases. Protein sequence databases provide comprehensive information on proteins (data on biological entities). Universal databases storing information on proteins from all organisms have to be distinguished from specialized databases storing information about specific families or groups of proteins or about the proteins in a specific species. Biological information is also often annotated to the sequences. Annotations like the biological function of a protein or the cluster a protein belongs to by a special classification are provided as texts (see above: “nucleotide sequence databases”).

Protein Structure Databases. Protein structure databases intend to make available the rapidly increasing number of known protein structures. The complete 3D structures of proteins are represented by storing the coordinates in a normed 3D space. Often textual annotations are provided, as usual. These databases also provide data on biological entities.

Mixed Type Databases. As mentioned above, some databases can be added to more than one of the given classes. Some database providers sometimes even offer more than one repository with different contents.

Interconnections between molecular biology databases

Many molecular biology databases provide links to each other. In general these cross-references are based on primary keys. Most databases provide unique accession numbers for each of their entry as primary keys. There are two kinds of interconnections between molecular biology databases:

- Hypertext links: Entries of several databases supply hypertext links to related databases (e.g. an entry in GenBank provides hypertext links to entries in EMBL Nucleotide Sequence Database or in SWISS-PROT).

- Textual references: Entries of several databases provide textual information about the accession number of the according entry in an other database. The user must perform the query for this accession number using the query interfaces of the other database.

It should be mentioned that hypertext links are usually implemented as textual references (especially in flat files). The database overview, as presented in Appendix A, also focuses on the interconnections between the according database and others. But we only investigate, whether there are hypertext links, or – if not – at least textual references.

Another classification which is interesting in this context addresses a biological relevant aspect of cross-references. We do not focus on this classification, but it should be mentioned here:

The links can be characterized into similarity links and other biological links.

Similarity links connect sequence entries or according data items with other items that provide or are based on similar sequences. Entries which are similar based on their sequence data are often called neighbors. Neighbors are usually identified using similarity search programs such as BLAST [4] or FASTA [56]. Usually, similarity links are not explicitly stored in molecular biology databases. They have to be computed by users utilizing similarity search programs.

Other biological links usually refer to additional biologically relevant information. E.g. an entry in SWISS-PROT providing information about a protein is linked to an entry in GenBank to refer to information about the gene encoding this protein.

3.2.2 Implementation Aspects: Data Modelling, Data Storage and Data Acquisition

After we got an overview on the data stored in molecular biology databases we now investigate how these data are modeled, organized and represented.

We use the following terminology. The logical structure of the data in a database – called schema – is specified by a data definition language (DDL). The transformation between the logical and internal level is operated by a data management system. Molecular biology databases use data management systems ranging from file management systems to database management systems (DBMSs)

This work will focus on exploring the schemas of the selected molecular biology databases and on the database management system, used for storing the schema physically on disk. We also investigate how data is acquired.

Data Modelling and Data Storage. Following [47] molecular biology databases can be classified in four categories according to the underlying data management system:

- Databases using a common DBMS
- Databases using ACEDB as a DBMS. ACEDB is a DBMS original implemented for the "A *C. elegans* Data Base"(ACeDB) but can be easily used for other biology resources (cf. 3.3.2).
- Databases using the Object-Protocol Model (OPM) on top of a common DBMS. OPM is a data model that combines standard object-oriented constructs with constructs for modelling scientific experiments (cf. 3.3.3).
- Databases implemented as Flat File Libraries. E.g. some of them use ANS.1[1] as DDL.

Detailed descriptions on the four different classes can be found in Section 3.3.

It is remarkable that most of the molecular biology databases were at first implemented as flat file libraries. But currently there are great changes in this field. Several flat file repositories are being re-implemented using standard relational or object-oriented DBMS. Due to this migration it should be emphasized once more that this compilation may be soon partly out of date.

Data Acquisition. An important aspect of databases in general is how new data are acquired. This aspect is closely related to the data modelling and storage. New data have to be inserted in the database with respect to the database schema. All DBMSs therefore provide their DDLs and their Data Manipulation Languages (DML).

Molecular biology databases can be classified in the context of data acquisition as follows:

- Data are acquired from other databases. This indicates that the format or model of the incoming data eventually has to be transformed.
- Data are acquired from the research community (all researchers who submit their data to the database). Sometimes, the right to submit data is restricted to a set of laboratories (e.g. the GDB acquires data only from laboratories that are members of the Human Genome Project). Data from lab submissions often contain errors which would require a manual or automatic check. Fill in forms make sure that the data fit to the schema of the database.
- Data are acquired from the literature. Usually this acquisition is done manually and requires a great contingent of well trained persons.

The frequency of update is another interesting question in this context. The frequencies of updates of molecular biology databases greatly vary. In fact, there are databases which are no longer maintained and updated, e.g. due to financial problems.

3.2.3 Implementation Aspects: Data Retrieval/Query Answering

The external level of a database is the interface to the user. This interface provides different views for different groups of users as well as a direct way for retrieving and manipulating the data with a data manipulation language including the DDL and a query language (QL). Thus the external level fixes the way how the data is represented, i.e. how the user can query against the database.

[1] ANS.1 (Abstract Syntax Notation No. 1) is a formal notation originally used for describing data transmitted by telecommunications protocol. It is a standard that defines a formalism for the specification of abstract data types [5]

Query Interfaces. According to [47] there are two different kinds of query interfaces:

- Fixed-form/Canned queries:
  The query form provides a view on the database. Queries can only be performed against a predetermined set of tables, classes or other database components and a predetermined set of attributes for each database component. The view may not necessarily reflect the underlying structure of the database.

- Free-form/*Ad hoc* queries:
  The query form does not have any predetermined structure. Usually these queries are expressed in some QL, depending on the underlying data model and DBMS. Although these QLs are often very powerful they have one big disadvantage: the user must have some detailed knowledge about the data model and schema of the database, of the particular QL and also some further skills in database theory. To formulate a query in SQL for example requires knowledge of the structure and manipulation of relational databases. Most biologists accessing molecular biology databases do not have these skills.

Most of the databases provide both kind of query interfaces. In this work we investigate the query interfaces of the selected databases. We focus on whether there are fixed- or free-form queries and how they are supported, how the query-interfaces are documented and if existing what QLs are supported.

Other ways of data retrieval. Often other possible and common ways of data retrieval in addition to the query interfaces on the according web-sites are the following:

- Download the data in flat file(s) via FTP. The capacity of one flat file often ranges from the whole database contents to single database entries.

- Several molecular biology database providers have classified their data to build up hierarchies. These hierarchies can then be browsed through to retrieve special data items. This way of data retrieval is often called *indirect data retrieval*. It should be mentioned that the classification criteria are often useful for the user, but can also be hindering.

We do not focus on the ways of data retrieval and query answering mentioned in this paragraph. But we also try not to survey them to make this work as complete as possible.

3.3 A Computer Scientist's View of Molecular Biology Databases

3.3.1 Standard Database Management Systems

Many molecular biology databases are implemented using a standard (often commercial) DBMS. The spectrum includes relational, object-oriented and object-relational systems.

Relational DBMS. Commercial relational DBMSs rely on a data model based on tables (also called relations) and flat tuples. A tuple is an ordered collection of attribute values. A relational DBMS based on the relational model does not provide (object) types nor collections as attribute values, nor inheritance over class-subclass hierarchies. Relational DBMSs support (a variant of) SQL (developed

1974 at the IBM Almaden Research Laboratory, San Jose, California by Chamberlin D. et al.) as DDL and QL.

Relational DBMSs are nowadays becoming more widespread also in the field of molecular biology databases. More and more databases with different contents use relational a DBMS. For example the Genome Sequence Database (GSDB, cf. A.38), a nucleotide sequence database, is implemented using the relational DBMS Sybase. CyanoBase (cf. A.17), a genetic database, is also implemented using the relational DBMS Sybase and the proteomic database YPD (cf. A.119) is managed by the relational DBMS Oracle.

Object DBMS. Most commercial object DBMSs follow the ODMG standard [15] that specifies the following features:

1. An *object data model* that allows classes with attributes and relationships, class hierarchy with inheritance and a range of collection types such as set, bag (multi-set), and list (sequence).

2. A (descriptive) *query language* (OQL) that has a syntax and power similar to SQL but acts on the richer ODMG model.

3. A *collection of "language bindings"* typically for Java, C++ and Smalltalk. A language binding allows the programming language to query the database using the OQL.

For example the INTERACT database (cf. A.49) is implemented using the object DBMS Poet. One of the MIPS databases (c.f. A.63) is also managed by an object DBMS. Another molecular biology database using an object DBMS is PSD/PIR (cf. A.79).

Object-relational DBMS. Object-relational DBMSs are relational DBMSs extended with implementation for object and collection types as attributes of tuples as well as class hierarchy with inheritance. They usually support SQL or at least a comparable DDL and descriptive QL. Some vendors (e.g. Oracle) of relational DBMSs provide object-relational versions of their software. Currently several databases at the Protein Information Resource (PIR, cf. A.79) seem to be the only databases implemented using an object-relational DBMS. In addition, the PRINTS-S database (cf. A.82) is built upon an object-relational schema.

3.3.2 ACEDB : A Database Management System originally developed for a molecular biology database

ACEDB [23] is a DBMS which was originally developed and implemented by R. Durbin and J. Thierry-Mieg for the "A *C. elegans* Data Base" (ACeDB) but was extended to be used for other specialized databases. There are often some misunderstandings concerned with the name: Originally ACeDB is the specialized database for the *C. elegans* organism and ACEDB is the underlying DBMS of this database. Several papers and researchers mix the terms ACEDB and ACeDB with each other.

Data model. ACEDB is similar to an object-oriented system. Data are modeled as objects that are organized in classes. However, ACEDB supports no class hierarchies, nor inheritance.

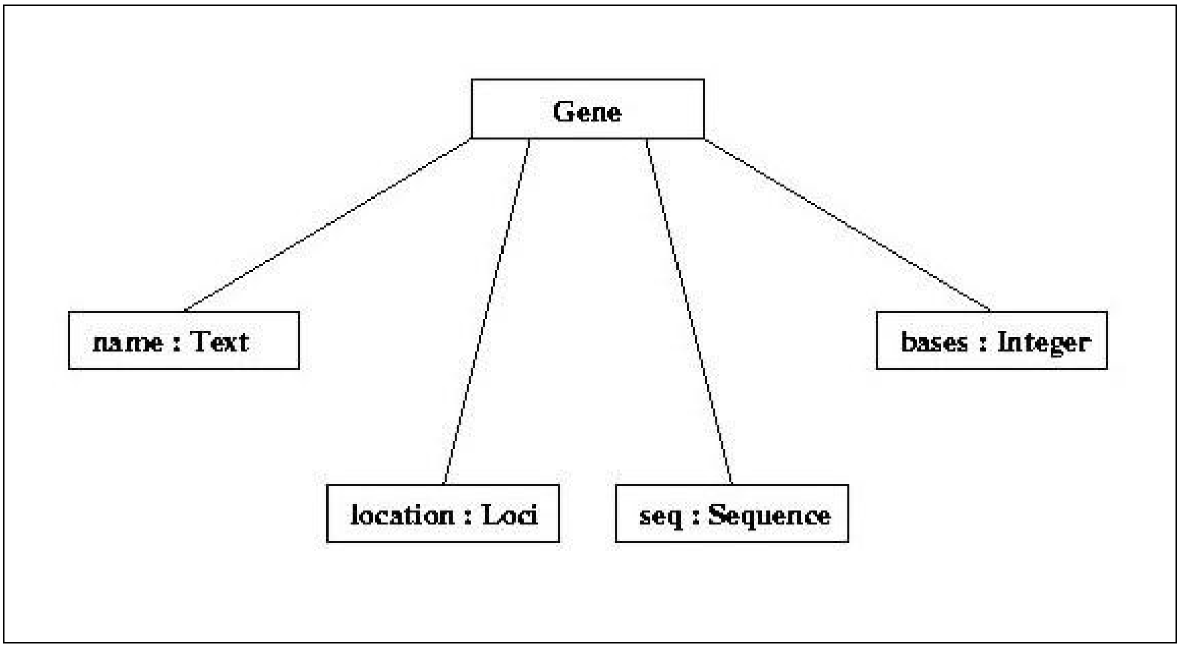


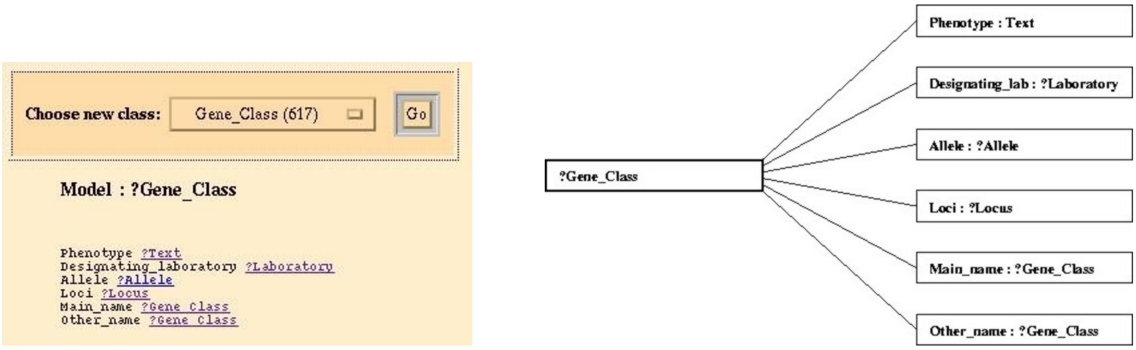
FIGURE 3.1: Exemplary model of an object class “Gene”.

Each object has a set of attributes organized as a tree. The nodes of the tree are all named. The branches typically model pointers to other objects or data with atomic (numerical and string) values. Objects are allocated to classes. Each class has a model (here called ”class-model”), specifying the maximum fan out of each node, and the type of data or classes of pointer permitted at each position. Individual instances of the class may have only a part of the branching pattern permitted by the model. This approach has several additional advantages:

- The modelling of sort of semi-structured data is supported. (One can compare an instance of a ACEDB class model with an XML-instance of an XML-DTD [2].)
- Extensions of the schema can be easily done by adding branches to the model. Such extensions do not require to restructure the whole data model because one need not to care about existing instances.
- Arbitrary (text) comments can be attached freely at any point in the tree. These comments are ignored by the searching algorithms of the ACEDB system. Comments allow flexible annotations.

The ACEDB approach to data modelling is very loosely related to data modelling with mark-up languages such as XML, i.e. to the so called semi-structured approach to data modelling [2]. However, the semi-structured approach offers also multiple inheritance which is not allowed in ACEDB (but it can be simulated by the ACEDB data model). Although in ACEDB the number of classes is restricted, a wider variety of objects within a class is allowed, so that two objects of the same class have only few or even no branches in common. E.g. in the ACeDB there are only about fifty classes that hold nearly a hundred thousand heterogeneous objects.

In addition to the object classes organized in trees there are more simpler classes which contain general arrays of data in ACEDB. These classes allow more rigid but more efficient storage of data such as



(A) Textual representation **(B)** Tree representation

FIGURE 3.2: Sample entry in the ACeDB database called "Gene_Class"

DNA sequences. They are organized as tables of records of variable length (tuples).

Implementation. ACEDB is a portable data management system with a powerful and extendable, graphical interface. It is available for several operating systems such as several UNIX machines, Windows and Macintosh. Via this interface there are several possibilities of querying an ACEDB database:

- **Template Search:** This is a fixed-form query interface where the user has to choose the class in which model to search.
- **Text Search:** The user has the possibility to search all text in all fields for a match to a text string.
- **Sample Queries:** Via a special fixed-form tool called "Query Builder" the user gets a more detailed view on the database. Target objects can be specified in more detail by properties of their attributes. A "`Where`" operator is used to apply criteria to a keyset (i.e. a set of primary keys, identifying a group of objects that are conform to desired criteria) that has previously been found.
- **Query Language Search:** ACEDB provides a powerful query language called AQL. Databases implemented using ACEDB can be searched via an *ad hoc* query interface using AQL, but that requires knowledge of AQL and of the data model and is often carried out slowly by the DBMS. The basic operations of AQL are:
  - to perform filtering operations on a set of objects, either on the basis of their names or the data they contain, and
  - to follow pointers to retrieve other objects.

  These operations can be combined using boolean operations.
- **Combining Keysets:** As mentioned above, a keyset is a set of primary keys, identifying a group of objects that conform to desired criteria. The result of queries produce different keysets

which can be saved. After collecting a number of keysets by querying the database the user can view commonalities or differences between them. A set of options to combine keysets provide set operators including intersection, union, union but not intersection (exclusive union) and difference of keysets.

Basic principles of DBMSs such as transaction, crash recovery and indexing are supported by ACEDB. The storage of data on disk is done according to the classes. Thus the tree objects are stored in a very different way from how the long stretches of DNA sequences are stored. The ACEDB implementors claim that this concepts make the system to be efficient.

ACEDB offers several analysis tools via its graphical interface but additional provides output in flat-files following several standard formats such as FASTA-format (special format for the FASTA analysis tool) or ANS.1 format recommended by the NCBI (NCBI provides several tools for biological data analysis which require input in a special format). The source code of ACEDB is public and every database developer is thus able to add own tools and features.

The ACEDB software can be downloaded at several URLs, e.g. at [3].

A comprehensive list of all databases that use ACEDB as DBMS is hard to obtain and maintain ([ACEDOC]). Especially genetic databases which provide specialized information on one organism often use ACEDB as their DBMS. [3] provides an incomplete list of molecular biology databases implemented using ACEDB. A very extensive collection of documentation on all aspects of ACEDB is available at the same URL. Vicarious as an example the "A *C. elegans* Database"(also known as ACeDB) should be mentioned: it is *the* database for which the ACEDB software has been primarily developed.

3.3.3 OPM: The Object-Protocol Model (OPM)

The Object Protocol Model (OPM) was introduced by [16] in 1995. It was developed for modelling of biological data, focussing on the modelling of scientific experiments. It is a data model that is very similar to an object model but provides specific constructs to support the modelling of scientific experiments. A suite of OPM data management tools provides an interface to common relational DBMSs like Sybase or Oracle. These tools include:

- a graphical schema editor for specifying OPM schemas;
- a translator of OPM schemas into relational definitions and procedures;
- a generic graphical querying, browsing and data entry tool with a WWW interface, for entering OPM data into an OPM-based database, specifying OPM queries and browsing their results;
- a tool to determine the OPM schema definition of existing relational database schemas.

Modelling of scientific experiments is often very important in the context of molecular biology databases because many users want to get information on the experiments which produced the data they are searching for. For example if one found a special nucleotide sequence in a database it is interesting by which methods this sequence was determined and by which methods the additional informations were produced.

Motivation for the development of OPM. [16] stated that modelling of scientific experiments with the constructs of a relational or object model is inadequate. To check this statement we will take a closer look on scientific experiments and their particularities.

Scientific experiments (often also called protocols) are instances of procedures such as genome sequencing protocols performed in molecular biology laboratories [16] and characterized by properties like time and location. Additionally experiments

- transform some input (resources) into output (experimental results);
- can be preceeded or succeeded by other experiments;
- can be components of more generic experiments or can consist of sequences of component experiments and/or alternative component experiments, where experiments and according component experiments share common inputs and/or outputs.

We take a look on an example (from [16]):

In Section 2.6.2 we were investigating a biological application called sequencing method. This experiment regards the construction of a genetic map (here called *contig map*) to determine the complete DNA sequence of a special organism. But the sequencing methods only allow sequencing of fragments of a few hundred nucleotides. Thus the chromosomal DNA has to be cut in smaller fragments, the fragments have to be sequenced and then assembled into contig maps.

These contig maps can be constructed out of sequenced DNA fragment using a protocol called `Construct`. The `Construct` protocol has an input (DNA `fragments`) and an output (`contig maps`) and can be carried out by two alternative protocols, `Overlap` and `Constraint`, both followed by protocol `Assemble`. Figure 3.3 shows the experiment `Construct` diagrammatically. Protocol `Overlap` regards the comparison of two DNA fragments using a computer program, while protocol `Constraint` regards the manual comparison of two DNA fragments according to certain constraints.

With the help of this example we now investigate why scientific experiments cannot be modeled adequately with the constructs supported by object data models according to [16]. For instance, the relationship between a generic experiment (e.g. protocol `Construct`) and its component experiments (e.g. protocol `Overlap`) cannot be modeled as a class-subclass relationship (`Construct` can also be carried out by subprotocol `Constraint` thus some instances of `Construct` are not even related to `Overlap` instances).

It is also inadequate to model the relationship between a generic experiment and its component experiments using attributes. We suppose, e.g., that `Overlap`, `Constraint`, and `Assemble` are modeled as three attributes of `Construct`. Then it is not clear that `Construct` can be carried out by alternative protocols `Overlap` and `Constraint`.

In this particular example, subprotocols of `Construct` can be modeled using a tuple attribute such as

```
attribute steps(step1,step2) :  (Overlap or Constraint, Assemble)
```

if the underlying data model supports both tuple attributes and union value classes.

However, such representations are not appropriate for more complex subprotocol structures and cannot capture the input and output relationships of experiments.

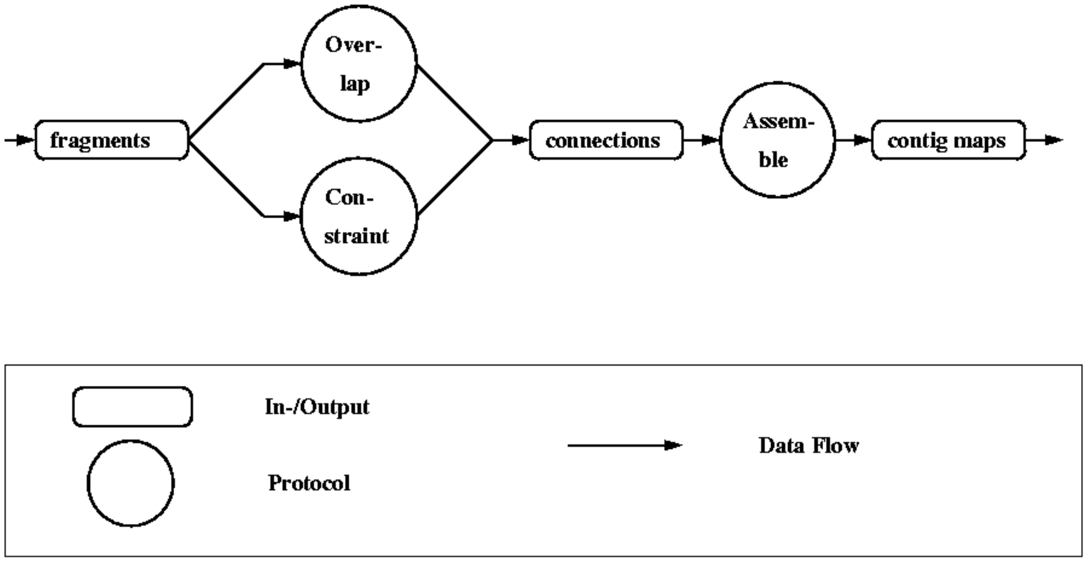


FIGURE 3.3: Example Protocol `Construct`: The `Construct` protocol can be carried out by two alternative protocols, `Overlap` and `Constraint`, both followed by protocol `Assemble` (taken from [16]).

Protocol connections and recursive specifications can be simulated in object DBMSs using methods but this involves manually developing numerous procedures in a programming language. Such procedures are hard to comprehend, time-consuming to develop and not easy to be exported to other DBMSs [16].

OPM supports the accurate modelling of scientific experiments, the specification of views, an underlying ontology of scientific terms, and nested queries prevalent in scientific applications.

Data model. OPM integrates constructs for modelling scientific experiments into an object data model.

The object constructs of OPM are similar to the so-called Semantic Database Model (SDM) and O_2 including object and derived object classes as well as inheritance mechanisms (see [16] for a detailed and more formal description).

Additional OPM provides constructs for modelling experimental procedures:

- **Protocol Classes** are similar to object classes. Protocol modelling is characterized by the recursive specification of generic protocols in terms of component subprotocols. These components can be
  - alternative subprotocols,
  - a sequence of subprotocols, and
  - optional subprotocols.

  All this can be expressed adequately with OPM.

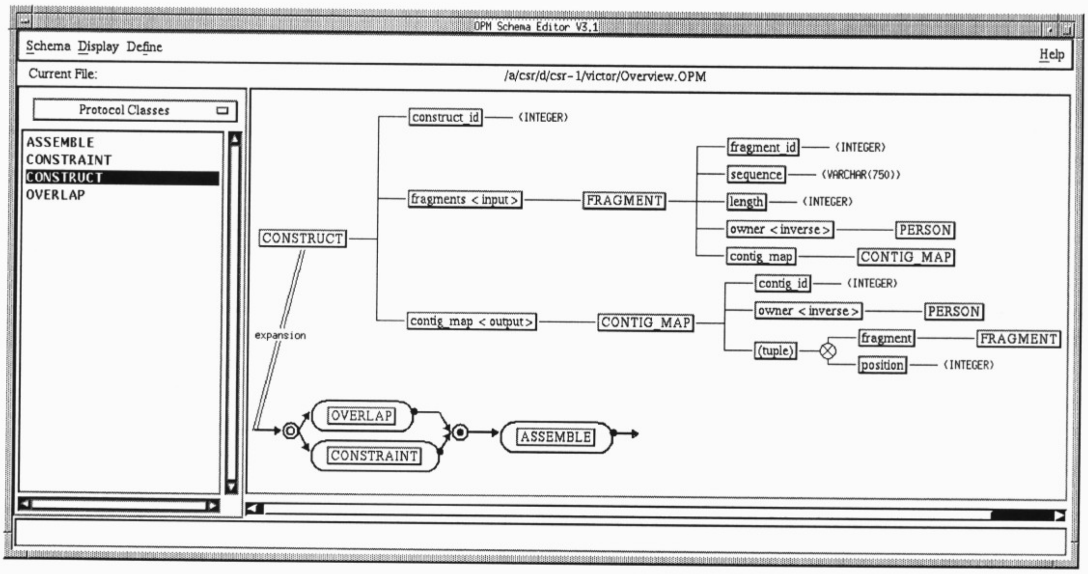


FIGURE 3.4: Graphical representation of the schema of the example protocol `Construct` (taken from [16])

- **Input and Output Attributes** are associated with a protocol class in addition to regular and connection attributes. They represent input and output data and are used to express input-output connections of directly related protocols.
- **Protocol relationships** are expressed using delete rules associated with the connection and system attributes.
- **Derived protocol classes** can be
  - generic protocol classes for representing experiments that are constructed from instances of existing protocol classes, and
  - subprotocol classes for representing subsets of experiments or a component of an existing experiment; a derived subprotocol inherits attributes from its generic protocol.

OPM provides constructs for expressing queries that are prevalent in scientific applications [16]. The OPM query language has a SQL-like structure and supports path expressions and set predicates. Other object oriented query constructs are not yet being implemented.

OPM currently works on top of relational DBMSs and makes use of all the advantages of their well investigated systems. As many scientific and molecular biology databases nowadays use object DBMSs, a new version of OPM and its tools is being developed to be suitable for that kind of systems in the future. A special feature will be the modelling of protocol connections with methods. The problem mentioned above is going to be carried out by the OPM tools so that the methods are generated automatically.

For example the Genome Data Base (GDB) at John Hopkins School of Medicine in Baltimore/USA is modeled and implemented using OPM on top of the relational DBMS Sybase. Together with the new version of the Protein Databank (PDB) at Brookhaven, GDB is one of the first databases to use OPM. In addition, several commercial biotechnology companies seem to use OPM for their databases.

3.3.4 Flat File Repositories

In the early days of molecular biology databases DBMSs were rarely used. Instead most molecular biology databases were built up using (more or less) indexed (flat) files. In an indexed flat file each entry consists of line types which are indexed with a keyword.

Even later in the eighties and nineties when (especially relational) DBMSs became more widespread, molecular biology databases followed this evolution only slowly because most molecular biology databases were administrated by biologists and the overhead costs for porting data to (commercial) DBMSs would have been very high. Another reason for this may be that the storage of large sequence data in standard DBMS seems to be not very efficient.

```
Publications = {[title: string,
        authors: {|[name: string,initial: string]|},
        journal: <uncontrolled: string,
              controlled: <medline-jta: string,     % Medline journal title abbreviation
                          iso-jta: string,          % ISO journal title abbreviation
                          journal-title: string,    % Full journal title
                          issn: string>>            % ISSN number
        volume: string,
        issue: string,
        year: int,
        pages: string,
        abstract: string,
        keywd: {string}]}
```

Description	Notation	ASN.1 terminology
list	$\{\|\tau\|\}$	sequence of
set	$\{\tau\}$	set of
record	$[l_1 : \tau_1, \ldots, l_n : \tau_n]$ (labeled fields)	sequence
variant	$<l_1 : \tau_1, \ldots, l_n : \tau_n>$ (tagged union)	choice

FIGURE 3.5: Part of a typical example of a complex type in ASN.1 data sources (taken from [20])

Nowadays several databases originally implemented as flat files are rebuilt using a standard (i.e. relational, object, or object-relational) DBMS. Note that there are still some databases which are based upon just one or a few indexed flat files.

One can argue that DBMSs might be dispensable because data entries (often) are not expected to change, multiple-user access is (often) not required, and the porting costs mentioned above are (often) to high. Another very important issue is that the data are often very complex. An example of a complex type is shown in Figure 3.5. Note the nesting of a set of keywords (strings) within the `keywd` record field of the type `Publication`, the nesting of a list of author records within the `authors` record field, and the use of a variant type within the `journal` field. Summarizing, typical data types used in theses (flat file) formats include deeply nested records, sets, lists and variants. These data can not easily be represented in a relational DBMS [20]. Object-oriented database products, which might be adequate to conquer these modelling challenges accurately, did not at the point of time (when molecular biology databases were established) adapt well to schema evolutions. Thus the need to model complex objects in a system which could adopt well to schema evolution led to the development of expressive formats such as ANS.1 and ACEDB.

Even if the implementation uses a standard DBMS (with or without OPM) or ACEDB, many databases offer a flat file release of the database. The reason for this is that flat files are a *de facto* data exchange standard in molecular biology data analysis. Biology researchers are used to several useful and sophisticated computer programs such as FASTA [4] or BLAST [56] – sequence analysis and retrieval tools – which work on (and often require a special format of) flat files. Thus, they often download data from molecular biology databases to analyse it on their local work stations. Biologists write their own parsing and analysing tools, which is of course not very reasonable, because this is time consuming and the programs can easily be faulty, but simply traditional. Moreover, biology researchers might need a tool that does not exist yet or is not available, so that they try to implement it for their own. E.g. several laboratories have their own parsing tools for extracting the data they are interested in out of flat files.

Another reason for the use of flat files as data exchange format might be that searching and accessing data via the web can be time consuming. Several biologists periodically download the current version of entire databases plus data retrieval tools to have a local copy. Access to the local data is often much faster than via the web.

Data modelling. Molecular biology databases implemented as flat files often do not have a data model *per se*. But most of them provide search indexes. Each data entry is structured by using keywords as indexes. The term ”line type” is often used for this construct. The keywords are often two-character strings but can be also whole words of variable length. If the data consist of complex types, these types will be trimmed to fit in the line types. Figure 3.6 shows a sample entry of the SWISS-PROT database (protein sequence database, cf. Appendix A.103). Every line begins with a two-character keyword. Some indexes are only allowed to appear once in an entry, others may appear more than once or may miss at all.

There are also molecular biology databases that do not provide indexing (e.g. HDB, cf. Appendix A.40). In these databases the data items are usually implicitly structured (see Figure 3.7)

Especially sequence databases are often flat file repositories, for modelling and storing a long sequence of nucleotides or amino acids is not jet investigated accurately.

Some databases (for example the well known database GenBank, cf. Appendix A.36) use ANS.1 to define the structure of their data items. ANS.1 (Abstract Syntax Notation No. 1) is a formal notation originally used for describing data transmitted by telecommunications protocol. It is a standard that defines a formalism for the specification of abstract data types [5]. Figure 3.5 shows the ASN.1 definition of a complex example type called `Publication`.

It seemed to be that there is no common semantic structure in all flat file databases. The keywords and indices of different databases may differ not only syntactically but also semantically. There is no information available about a central organization that identifies a set of standard line types for each flat file entry.

On the other hand, the three databases of the IC (International Nucleotide Sequence Database Collaboration) — GenBank (cf. Appendix A.36), EMBL (cf. Appendix A.28), and DDBJ (cf. Appendix

```
ID   PILI_PSEAE     STANDARD;      PRT;   178 AA.
AC   P43502;
DT   01-NOV-1995 (Rel. 32, Created)
DT   01-NOV-1995 (Rel. 32, Last sequence update)
DT   30-MAY-2000 (Rel. 39, Last annotation update)
DE   PILI PROTEIN.
GN   PILI.
OS   Pseudomonas aeruginosa.
OC   Bacteria; Proteobacteria; gamma subdivision; Pseudomonadaceae;
OC   Pseudomonas.
RN   [1]
RP   SEQUENCE FROM N.A.
RC   STRAIN=ATCC 15692 / PA01;
RX   MEDLINE=94195101; PUBMED=7908398;
RA   Darzins A.;
RT   "Characterization of a Pseudomonas aeruginosa gene cluster involved
RT   in pilus biosynthesis and twitching motility: sequence similarity to
RT   the chemotaxis proteins of enterics and the gliding bacterium
RT   Myxococcus xanthus.";
RL   Mol. Microbiol. 11:137-153(1994).
CC   -!- FUNCTION: MAY BE A PART OF A SIGNAL-TRANSDUCTION SYSTEM THAT
CC       REGULATES TWITCHING MOTILITY BY CONTROLLING PILUS FUNCTION
CC       (EXTENSION AND RETRACTION).
CC   -!- SIMILARITY: CONTAINS 1 CHEW DOMAIN.
CC   --------------------------------------------------------------------------
CC   This SWISS-PROT entry is copyright. It is produced through a collaboration
CC   between  the Swiss Institute of Bioinformatics  and the  EMBL outstation -
CC   the European Bioinformatics Institute.  There are no  restrictions on  its
CC   use  by  non-profit  institutions as long  as its content  is  in  no  way
CC   modified and this statement is not removed.  Usage  by  and for commercial
CC   entities requires a license agreement (See http://www.isb-sib.ch/announce/
CC   or send an email to license@isb-sib.ch).
CC   --------------------------------------------------------------------------
DR   EMBL; L22036; AAA25951.1; -.
DR   INTERPRO; IPR002545; -.
DR   PFAM; PF01584; CheW; 1.
KW   Sensory transduction.
SQ   SEQUENCE   178 AA;  19934 MW;  634A1A4B135A7E77 CRC64;
     MSDVQTPFQL LVDIDQRCRR LAAGLPAQQE AVQSWSGIGF RMGGRFFVAP MGEVGEVLHE
     PRYTQLPGVK TWVKGVANVR GRLLPIMDLC GFLGTELSPL RKQRRVLVVE HLDVFAGLIV
     DEVFGMQHFP VDTFSEQLPP LEAALQPFIH GVFHREQPWL VFSPHALAQH QGFLDVAV
//
```

FIGURE 3.6: SWISS-PROT database: A typical entry of a flat file combining various kind of data

A.22) — seemingly have a set of syntactically different but semantically equal line types. This might be reasonable because the contents of the three databases are very similar.

Databases with different contents have more differences. The semantic of the annotated information is often already encoded in the keyword. As e.g. entries of GenBank and SWISS-PROT store different data, the semantic structure of the entries of databases storing different biological data (belonging to different classes according to the classification by contents) is less similar.

A well known example of the important role that indexed flat files still play in the field of molecular biology databases is the IC. The exchange of data between GenBank, EMBL, and DDBJ is carried out using formated flat files. As the entries of the different databases have different structures (e.g. the keywords of GenBank have different length and the keyword in EMBL consists of two characters), and semantically different indexes, complex parsing tools carry out the format conversions. It should be emphasized that full conversion often cannot be done by these tools.

3.3.5 Summary

Biological data are complex.

In general, knowledge bases in molecular biology must capture three types of data (see also [57]):

```
>HSPM3|HSPM3 histone H3 - garden pea. [ Pisum sativum ]|peaH3
ARTKQTARKSTGGKAPRKQLATKAARKSAPATGGVKKPHRFRPGTVALREIRKYQKSTEL
LIRKLPFQRLVREIAQDFKTDLRFQSSAVSALQEAAEAYLVGLFEDTNLCAIHAKRVTIM
PKDIQLARRIRGERA

>HSPM4|HSPM4 histone H4 - garden pea. [ Pisum sativum ]|peaH4
SGRGKGGKGLGKGGAKRHRKVLRDNIQGITKPAIRRLARRGGVKRISGLIYEETRGVLKI
FLENVIRDAVTYTEHARRKTVTAMDVVYALKRQGRTLYGFGG

>HSPG4|HSPG4 histone H4 - pig. [ Sus scrofa domestica ]|porH4
SGRGKGGKGLGKGGAKRHRKVLRDNIQGITKPAIRRLARRGGVKRISGLIYEETRGVLKV
FLENVIRDAVTYTEHAKRKTVTAMDVVYALKRQGRTLYGFGG
```

FIGURE 3.7: Sample entry of HDB: the data items are not structured explicitly using indexes.

1. Data on biological entities such as nucleic acids, proteins, etc. and on relationships between these entities (descriptive knowledge).

2. Data on the dynamic behavior of biological processes that consit of these relationships (behavioral knowledge).

3. Data on the methods which can be used to identify the entities and their relationships (dynamic knowledge).

A typical data type has a nested structure which is not easy to represent in the relational model. Object models are more adequate to model these complex data accurately (in [62] Schmelzer *et al.* state that even object models have to be extended by knowledge models originated from the AI field to model special particularities of biological data).

But in the beginning of molecular biology databases, object DBMSs did not adapt well to schema evolutions. Relational DBMSs often provided an unintuitive, fragmented design. Thus, formats to model these complex data accurately providing the ability to adapt to schema evolution have been developed. Such formats are e.g. ASN.1 and ACEDB. There were additional reasons to use formatted files, e.g. the easy access of those files from Fortran or C and the great overhead costs of commercial DBMSs.

Later, object DBMSs became more sophisticated, and several molecular biology database providers decided to invest in commercial (object and relational) DBMSs. Other molecular biology databases (especially the major data sources of the Human Genome Project [20]) have been using the object front-end OPM for their relational databases to create more intuitive designs.

From a computer scientist's point of view the two models of OPM and ACEDB are especially interesting. Both models provide constructs that seem to be applicable to other application domains than biology:

- ACEDB is a sort of semi-structured approach as noticed in [2]. Moreover, this approach has become widespread in the field of molecular biology data modelling and management.

- OPM models time dependent procedures. The model might be very useful in or related to several other application areas, e.g. for modelling of document-based workflow. An interesting question is whether metabolic pathways (which provide a time dependent aspect) can be modeled (adequately) with OPM. This is investigated in Chapter 5.

Another aspect which is especially interesting for computer scientists is the fact that most molecular biology databases rely on several different models. The heterogeneity of all these data stored in different databases with different formats needs integration. Data integration is an up-to-data research issue in the field of federated database systems. Molecular biology databases are supposed to be a great application domain for this computer science research field. See Chapter 4 for an overview over this issue.

Finally it can be stated, that the research of molecular biology databases is extremely up-to-date and very interesting from the computer scientist's point of view, of course especially for researchers dealing with modelling of semi-structured data, structured texts and data integration of heterogeneous database systems.

3.4 A (Tentative) Biologist's View of Molecular Biology Databases

While a computer scientist's view of molecular biology databases focuses on how they are structured and managed, a biologist's view is primarily concerned with their use.

As it was already stated above, the amount of data is too large to keep it within one single brain. Therefore, molecular biology databases and their associated tools are a key technology for molecular biology. With the help of the available information and analysis tools, biologists save time and expenses in their daily research. (Example: Several proteins of the fully sequenced genome of yeast are not yet investigated, that means, their sequences are known, but their functions are not. If a researcher wants to determine the biological function of such a protein, there are several pieces of data in several databases, pointing to the putative function. These hints are usually not proven. This is the task of the researcher who wants to determine the function of the protein. But the researcher has already a clue what he is most likely looking for.)

Nevertheless, the biologists' use of molecular biology databases is handicapped by several obstacles. We try to list some major of these obstacles in the following:

No clarity. The domain of molecular biology databases is a rapidly evolving and increasing area. Also for biologists, it is difficult and perhaps even impossible to keep track of the available data sources. Many biologists usually use a small set of three to ten molecular biology databases for their daily work, according to their research foci and the foci of the databases. For example a researcher investigating proteins in yeast, will obtain information from SGD (cf. Appendix A.100), YPD (cf. Appendix A.119), and MIPS/Yeast (cf. Appendix A.63). As a consequence, biologists do not have all information that might be important for them at their disposal.

Query interfaces. In general, most biologists use fixed-form query interfaces or browsable hierarchies to retrieve data. Only few provide knowledge in querying a database via an *ad hoc* query interface using a query language. Although these fixed-form query interfaces often only provide a limited view on the database, many biologists are happy with them. But different databases offer different query interfaces. Biologists often do not have the time to learn the ropes of several different query interfaces.

Analysis tools. Analysis tools are a key to biologists to interprete and understand data from molecular biology databases. Usually these analysis tools work on the sequence data. They predict e.g. opening reading frames (ORFs) in nucleotide sequences or secondary structure motifs of amino acid sequences.

In fact, there is a large amount of sophisticated analysis tools for molecular biology data available via the web. But most of them have the same functionality. There are only few tasks that can be worked out by a large set of different tools.

Most biologists have their own preferred tools for each task. For example, one might always use the BLAST [4] programs for sequence similarity search and never the FASTA [56] program. Some tools may also produce inadequate results, because they are based on unsuitable algorithms.

Of course, biologists are not familiar to the underlying algorithms. They need some practical experience to decide which tools are qualified, and which are not.

Another problem of analysis tools for biological data is the following: Many tools are complex to use because they have several input parameter that influence the results intensively, but are hard to manipulate without any knowledge about the theoretical basics of the underlying algorithms. In addition, they are sometimes poorly documented, if at all.

As a consequence, most biologists use these tools with their suggested standard parameters and have never modified any input parameter. This situation limits the use that can be made of these tools.

Literature links. Most molecular biology databases offer literature references. This is really needed, because biologists are used to retrieve information, additional to the contents of the databases, from the literature. But often, the literature links provided by molecular biology databases are not up-to-date. Indeed, usually literature references have the actuality of their corresponding data entries and are not updated later. These references often lead to articles that describe the same information stored in the databases.

As a consequence, biologists are interested in more up-to-date articles that usually describe new and additional information on the topic of interest. To get more up to date literature on a specific topic, biologists have to query bibliographic databases, such as PubMed (cf. Appendix A.90).

Because biologists still publish and retrieve a lot of their data only in/from the scientific literature, this problem of up-to-date literature links and references is, by all means, serious and time consuming.

Conclusion

Usually, biologists are no computer (science) experts. Most biologists want to query easy-to-use, standardized interfaces: they want to access, retrieve, and analyse biological data by “point-and-click” navigation without having to care about query languages, input parameters, etc.

A great benefit would be an integrated environment for molecular biology databases and associated analysis tools (see also Chapter 4). This would help to overcome the obstacles mentioned above, that keep biologists from using molecular biology databases in an optimal manner:

- No clarity: A transparent access to molecular biology databases would ensure that users perform their queries against all available information and do not have to identify the sources relevant for their queries.

- Query interfaces: A standardized, powerful query interface, providing fixed-form interfaces, *ad hoc* interfaces, browsable hierarchies, and knowledge-driven interfaces is required. An only slightly evolving interface would most likely attract biologists to become acquainted with the particularities of this interface.

- Analysis tools: A broad range of standard analysis tools would be required. Analysis tools, that seemingly do not yield adequate results should be left out. The use of these tools should be assisted by expert documentation or transparent interfaces.

- Literature links: An integrated information infrastructure should also include a bibliographic database (such as PubMed) to ensure that the newest articles are referred to and can be accessed.

Beside all efforts, to develop comprehensive databases with powerful query interfaces and sophisticated analysis tools, the requirements, characteristics, and preferences of biologists (the users!) should not be forgotten. There is no common abstraction level between biologists and computer scientists, and this makes communication between biologists and computer scientists very difficult.

3.5 Classification of Molecular Biology Databases: Grand Table

As mentioned above, we have listed 120 randomly selected molecular biology databases – investigated according to the classification criteria mentioned in 3.2 – in Appendix A.

This section summerises the results of classifying these 120 molecular biology databases. A tentative evaluation is presented in Section 3.5.1.

The results of the classification are listed in a grand table (cf. Section 3.5.2). Table 3.1 on page 48 provides the legend for this grand table.

3.5.1 Summary

An interesting observation is a certain kind of coherence between data model and contents of molecular biology databases. Of course, there are several exceptions to this coherence, but on the other hand there is a certain trend: Most genetic databases use the ACEDB software. Molecular biology databases providing annotated sequence data often use a flat file implementation. Databases on certain molecular biological interactions (e.g. protein interactions) often use object DBMSs. Relational DBMSs are used to manage all kinds of biological data.

3.5.2 Grand Table

In this section the results of the classification are listed. Each table item provides a page reference to the according entry in the database directory in Appendix A. The legend of the grand table is provided in Table 3.1.

TABLE 3.1: Legend of the grand table of molecular biology databases

Legend	
Database	short name of the database; in alphabetical order (digits before letter)
Contents	contents of the database
References	references (interconnections) to other databases: HT = hypertext links TR = textual references
Implementation	underlying data management system
Data Acquisition	C = research community D = other databases L = scientific literature
Querying/Retrieval	FF = fixed-form query interface AH = *ad hoc* query interface FTP = download data files (usually via FTP) Ind. = indirect data retrieval (browsing through hierarchies)
Details	page of this report where details about the database can be found

Database	Contents	References	Implementation	Data Acquisition	Querying/Retrieval				Details at page
					FF	AH	FTP	Ind.	
3DBase	prot. struct.	?	OPM	D	$\surd$				117
AAindex	mixed type	HT	flat file	L			$\surd$		118
AARSDB	nucl. sequ.	TR	flat file	L				$\surd$	119
ALFRED	genetic	HT	rel. DBMS	C, L	$\surd$			$\surd$	119
aMAZE	pathways	HT	obj. DBMS	D	under construction				120
AMmtDB	nucl. sequ.	HT	flat files	D	$\surd$	$\surd$	$\surd$		121
ASDB	genetic	HT	??	D	$\surd$				122
Axeldb	genomic	HT	ACEDB	C, D, L	$\surd$	$\surd$	$\surd$	$\surd$	123
BMRB	prot. struct.	HT	rel. DBMS	C, D, L	$\surd$		$\surd$	$\surd$	124
BRENDA	mixed type	HT	rel. DBMS	C, L	$\surd$				125
CATH	taxonomy	HT	??	D	$\surd$		$\surd$	$\surd$	126
COG	proteins	HT	flat files	D	$\surd$		$\surd$	$\surd$	127
Colibri	proteomic	HT	rel. DBMS	C, D	$\surd$	$\surd$	$\surd$		128
COMPEL	genetic	HT	rel. DBMS	D, L	$\surd$		$\surd$	$\surd$	130
CSNDB	pathways	HT	ACEDB	L	$\surd$	$\surd$	$\surd$	$\surd$	131
CyanoBase	genomic	HT	rel. DBMS	C,D	$\surd$		$\surd$	$\surd$	132
DAtA	proteomic	HT	rel. DBMS	D	$\surd$		$\surd$	$\surd$	133
DBcat	literature	HT	flat file	C, D, L	$\surd$	$\surd$	$\surd$	$\surd$	134

Legend on page 48

Database	Contents	References	Implementation	Data Acquisition	Querying/Retrival				Details
					FF	AH	FTP	Ind.	at page
dbSNP	nucl. sequ.	HT	rel. DBMS	C, D	√		√		136
DDBJ	nucl. sequ.	HT	rel. DBMS	C, D	√		√		137
DIP	proteins	HT	rel. DBMS	L	√		√		138
DSMP	prot. struct.	HT	flat files	D	√	√			139
EcoCyc	met. pathw.	**??**	obj. DBMS	D, L	√	√	√	√	141
EcoGene	proteomic	HT	flat files	–	√		√		142
EID	genomic	TR	flat files	D			√		144
EMBL	nucl. sequ.	HT	rel. DBMS	C, D	√		√	√	145
EMGLib	nucleotide sequ.	HT	flat file	D	√		√		146
ENZYME	mixed type	HT	flat file	C, D, L	√		√	√	148
EPD	genetic	HT	flat file	**??**	√		√		149
ExInt	genomic	HT	flat file	D	√		√		150
FIMM	mixed type	HT	flat file	D, L	√				151
FlyBase	genomic	HT	rel.DBMS	C, D, L	√		√	√	152
GDB	genomic	HT	OPM	C	√	√		√	154
GenBank	nucl. sequ.	HT	rel. DBMS	C, D	√		√		155
GIMS	genomic	**??**	obj. DBMS	D	under construction				157
GSDB	nucleotide sequ.	HT	rel. DBMS	D	√	√	√		159

Legend on page 48

Database	Contents	References	Implementation	Data Acquisition	Querying/Retrival				Details
					FF	AH	FTP	Ind.	at page
GXD	genomic	HT	rel. DBMS	C, L	$\surd$	$\surd$	$\surd$	$\surd$	162
HDB	mixed type	HT	flat file	D	$\surd$		$\surd$	$\surd$	163
HGBASE	genomic	HT	rel. DBMS	C, D, L	$\surd$	$\surd$	$\surd$		164
HGMD	genetic	HT	flat file ?	L	$\surd$				165
Hom. Res.	prot. sequ.	HT	rel. DBMS	D, L	$\surd$				166
HOX Pro	genetic	HT	rel. DBMS ?	**??**				$\surd$	167
IDB/IEDB	mixed type	HT	rel. DBMS	D			$\surd$		167
IMB	structure	HT	flat file ?	D	$\surd$			$\surd$	168
IMGT	nucleotide sequ.	HT	rel. DBMS	D	$\surd$	$\surd$	$\surd$	$\surd$	169
InBase	mixed type	HT	flat file ?	C				$\surd$	170
INTERACT	proteins	HT	obj. DBMS	C, D, L	$\surd$	$\surd$			171
InterPro	proteomic	HT	rel.DBMS	D	$\surd$		$\surd$		173
IXDB	genomic	HT	rel. DBMS	C, D, L	$\surd$			$\surd$	174
KEGG	met. pathways	HT	flat file	D	$\surd$		$\surd$	$\surd$	176
KinMutB.	mixed type	HT	flat file ?	C				$\surd$	179
KMDB	mixed type	HT	flat file	**??**			$\surd$	$\surd$	180
LIGAND	mixed type	HT	flat file	D, L	$\surd$		$\surd$	$\surd$	181
MAGEST	genomic	HT	rel. DBMS	D	$\surd$		$\surd$		181

Legend on page 48

Database	Contents	References	Implementation	Data Acquisition	Querying/Retrival				Details
					FF	AH	FTP	Ind.	at page
MaizeDB	genomic	–	rel. DBMS	??	√	√		√	182
MDDB	mixed type	??	rel. DBMS	??	??	??	√	??	183
MEROPS	taxonomy	HT	flat file	D	√		√	√	184
MGD	genomic	HT	rel. DBMS	C, D	√	√	√	√	185
MIPS	proteins	HT	diverse	C	√			√	186
MitBASE	genomic	HT	rel. DBMS	C, D, L	√			√	187
MitoNuc	genetic	HT	flat file	D, L	via SRS (cf. 4.3.7)				189
MITOP	proteomic	HT	flat file	D, L	√		√	√	189
MMDB	prot. struct.	HT	flat file	D	√		√		191
ModBase	prot. struct.	HT	flat file	D	√				192
MTB	genomic	HT	rel. DBMS	C, L	√				193
NDB	structure	HT	rel. DBMS	C, D, L	√			√	194
OMIM	genetic	HT	??	C	√		√		195
ooTFD	genetic	HT	obj. DBMS	D, L	√		√		196
ORDB	mixed type	HT	o-r. DBMS	C	√			√	198
PDB	prot. struct.	HT	rel. DBMS	C	√		√		198
PEDB	proteomic	HT	rel. DBMS	D, L	√			√	200
Pfam	prot. sequ.	HT	flat files	D	√		√	√	200

Legend on page 48

Database	Contents	References	Implementation	Data Acquisition	Querying/Retrival				Details
					FF	AH	FTP	Ind.	at page
PIR/PSD	prot. sequ.	HT	o.-r. DBMS	C, D, L	√		√		202
PLMItRNA	mixed type	TR	flat file	C, D, L	√				204
PombePD	proteomic	HT	rel. DBMS	C, L	√			√	205
PRINTS-S	taxonomy	HT	o.-r. DBMS	D	√	√	√	√	206
ProClass	taxonomy	**??**	rel. DBMS	D	√		√		207
ProDom	taxonomy	HT	flat file ?	D	√		√		208
PROSITE	proteomic	HT	flat file	D	√		√		209
ProTherm	mixed type	HT	flat file ?	**??**	√				210
ProtoMap	taxonomy	**??**	**??**	D	√			√	210
PseudoBase	genetic	HT	flat file	C				√	211
PubMed	literature	HT	**??**	**??**	√			√	212
RDP	nucl. sequ.	**??**	obj. DBMS	C, D			√		213
REBASE	mixed type	HT	**??**	D, C	√		√	√	214
RegulonDB	genetic	HT	rel. DBMS	D, L	√				215
RHdb	mixed type	HT	rel. DBMS	C	√		√		217
SacchDB	genetic	TR	ACEDB	C, L	√	√	√	√	218
SBASE	prot. sequ.	HT	flat files	D, L	√		√		218
SCOP	prot. struct.	HT	flat files	D	√		√	√	219

Legend on page 48

Database	**Contents**	**References**	**Implementation**	**Data Acquisition**	**Querying/Retrival**				**Details**
					FF	**AH**	**FTP**	**Ind.**	**at page**
SELEXdb	nucl. sequ.	–	flat file	L	via SRS (cf. 4.3.7)				220
SENTRA	mixed type	HT	flat file	D	√				221
SGD	genetic	HT	rel. DBMS	C, L	√	√	√	√	221
SMART	mixed type	HT	rel. DBMS	D	√				223
SRPDB	mixed type	–	flat file	D				√	224
SWISS-PROT	prot. sequ.	HT	flat file	D	√		√	√	225
SWISS-2dPAGE	proteomic	HT	flat files	**??**	√		√	√	229
TAIR	genetic	HT	obj. DBMS	C, D	√		√		229
TIGR	mixed type	HT	rel. DBMS	D	√		√		230
tmRDB	nucl. sequ.	HT	flat file	D, L				√	231
TRANSFC	genetic	HT	rel. DBMS	L	√	√	√	√	232
Transterm	genetic	–	rel. DBMS	D	√		√		233
TrEMBL	prot. sequ.	HT	flat files	D	√		√		214
TRIPLES	genetic	HT	rel. DBMS	C	√		√		234
TRRD	mixed type	HT	flat file	**??**				√	235
UK CropNet	genetic	HT	ACEDB	C	√	√	√	√	236
UTRdb	nucl. sequ.	HT	flat file	D	via SRS (cf. 4.3.7)				237
WIT	met. pathw.	HT	**??**	D	√				237

Legend on page 48

Database	Contents	References	Implementation	Data Acquisition	Querying/Retrival				Details
					FF	AH	FTP	Ind.	at page
WormPD	proteomic	HT	rel. DBMS	C, L	√			√	238
YIDB	genetic	HT	flat file ?	D, L				√	239
YPD	proteomic	HT	rel. DBMS	C, L	√			√	240
ZmDB	genetic	HT	ACEDB	C, D	√	√	√	√	241

Legend on page 48

Chapter 4

Molecular Biology Database Integration

The biology community is a distributed one. Each area generates its own data and information resources. Specialized tools to query and analyse these data are available but often associated with a particular database and tailored to its format.

Many biologists still query collections of stand-alone data resources. The incomparability and heterogeneity of these databases limits the use that can be made of the available knowledge. This is a great handicap, for biology is a knowledge-based discipline which, very often, makes use of existing data to make predictions and assumptions.

It is therefore very important to develop methods to integrate heterogeneous molecular databases and their associated tools, and to address the task of data fusion. There are many approaches that try to solve the problem of database integration in the field of molecular biology databases, ranging from theoretical approaches to running systems.

This part of the work is dedicated to investigate briefly the principles behind molecular biology database integration, especially the requirements and obstacles in the context of molecular biology database interoperability. A short but (hopefully) representative list of existing projects dealing with integration of molecular biology data sources is presented.

4.1 Introduction

"The biology community is a distributed one" [6]. Each area generates its own data and information source. As a consequence a range of specialist analysis and query tools is available, each typically associated with a particular database and its format.

Molecular biology databases have different structures, contents, and query languages. Analysis tools have no common interfaces and often work on a limited subset of the data. Usually these tools are complex to use and poorly documented.

Many biologists still query many different stand-alone resources. As a consequence, several non-trivial tasks – such as e.g. identifying relevant sources, communicate with and query different databases, transforming data from different formats, etc. – have to be carried out by the biologists. Additional, biologists have to handle several different user interfaces, each with its own capabilities of data retrieval. All this limits the use that can be made of the available information.

To address this problem there are many research groups working on the integration of molecular biology database.

Currently, molecular biology database interoperation is based on interactive navigation through hypertext links. Each molecular biology database tries to provide cross references to other databases. But – as [39] stated – “interoperation based on declarative queries to multiple network-accessible databases will support analysis and investigations that are orders of magnitude faster and more powerful than what the WWW supports using point-and-click navigation” .

As a consequence, there is a need for an integrated database infrastructure with powerful and declarative query capabilities.

4.1.1 System Requirements

A powerful system for molecular biology database interoperation is urgently needed to adequately make use of the existing knowledge in the domain of biology. But how should such an infrastructure look like? What are its requirements? What does the term “powerful” mean in this context? These questions are hard to answer, but we try to give a (possibly incomplete) list of system requirements for tools supporting the interoperation of molecular biology databases.

System Interaction. First of all, it can be stated that such a system is to be used by biologists. Following [39], this usage mainly consists of:

- general browsing: e.g. it should be required that the data is classified into hierarchies which can be browsed;
- seeking: particular information should be seekable through simple interfaces;
- complex querying: the user should be able to perform complex queries;
- analysing data: the system should provide sophisticated and standardized tools for (computational) data analysis;
- updates: biologists should be able to add newly produced data;

System Requirements. Ignoring the update problem, the previous thoughts on system interaction lead to the following (possibly incomplete) list of system requirements. A molecular biology database integration system should provide:

- advanced query capabilities including interfaces to express complex declarative multi-database queries, simple fixed-form query interfaces, and capabilities to browse the data;
- access to the newest data (updating of the member databases should occur about every day);
- transparent access, i.e. that users should not be required to know of the existence, the location, the access mechanism, and the contents of databases relevant to their query (but it could be helpful if users have a possibility to choose their favourite databases);

- transformation and merging of data from different source formats;
- a transparent schema, i.e. that users should not be required to be familiar with the schema of each component database they are querying;
- capabilities to visualize and analyse data in an adequate manner (this includes standard analysis and visualization tools);
- consistency on both levels of data modelling and data retrieval (see below);

4.1.2 Problems Arising with Heterogeneity

The major problems in molecular biology database integration are the different levels of heterogeneity and inconsistency. Molecular biology databases are created by such a diverse set of international groups. Nobody has the power to legislate standards at any single level of abstraction [39]. Such a legislation would be most likely inappropriate, for the field evolves very rapidly. There are new relationships between and new properties of biological entities constantly discovered. It is therefore simply impossible to define *the* schema of a specific biological entity or concept.

As a consequence, the existing levels of heterogeneity – such as conceptualization (e.g. a representation for genomic maps), data model (relational, object-oriented, ACEDB, etc.), or query language – will probably still continue to exist for a long time. Closely related are the problems which arise from the fact, that biological terms and notions are used in an inconsistent manner by different databases (heterogeneity at the semantic level).

Conceptual Heterogeneity. Molecular biology database providers frequently design different conceptualizations (i.e. models of the domain objects of interest with relevant attributes and relationships) of the same domain. For example the conceptualization of a “genomic map” may differ as follows. A genomic map may be represented allowing “loci” to exist in a total order, or in a partial order. Representations may allow loci to overlap or not. This is closely related to heterogeneity at the semantic level (see below).

Data Format Heterogeneity. As already often stated in this thesis, molecular biology databases differ remarkably according to their underlying data model and data management system. Data sources include standard DBMSs (relational, object-oriented) with SQL, OQL, or OPM interfaces, and structured text files in binary (ACEDB) or ASCII format on top of which indexing is often provided (flat file repositories) [20] (see also Chapter 3.3 and Appendix A).

Query Language Heterogeneity. Several databases have interfaces for their own (declarative) query language (e.g. SQL, OQL, AQL), even if they do not offer them to the user (e.g. some databases implemented using a relational DBMS do not offer an SQL interface for direct access via the public interfaces). But several databases do not provide any query language (especially flat file databases). This fact raises problems when thinking about an interface for complex declarative queries against all integrated databases.

Semantic Inconsistency. Naming conventions for data objects differ between databases. The concept *gene* is a prominent example. In GDB (cf. Appendix A.35) it is defined as a DNA fragment that can be transcribed and translated into a protein. For GenBank (cf. Appendix A.36) and GSDB (cf. Appendix A.38), however, a *gene* is a DNA fragment that carries a genetic trait or phenotype. This includes non-structural coding DNA regions like introns or promoters. It is clear that there is a distinct semantic difference between the two notions of *gene*. But both are used interchangeably causing misunderstandings and communication problems [63].

Another problem in this context is the complexity of the very notion of biological function. Biological function may be described at different levels, e.g. the molecular function or the cellular function of a protein. For an enzyme such as aspartokinase, the molecular function is the catalysis of a given reaction (characterised by several properties), whereas the documented cellular function of aspartokinase in bacteria is the catalysis of the first step in the common biosynthetic pathway (example taken from [71]). It becomes more complex because a protein with a given molecular function often is involved in very different cellular processes. Both cellular and molecular functions of a protein can also vary with its subcellular localization, the cell type in which it is expressed, or other physiological realities. The organization and representation of biological function in molecular biology databases reflect the databases' primary foci. It is not uncommon that e.g. the molecular function of an entry is described in the field or table "biological function", but the cellular function is hidden in another field or table (e.g. named "comment") or misses at all.

Summarizing, the problem of semantic inconsistency is that there are several biological concepts that could have several different definitions, depending on the perspective of the user. Each definition makes sense for different groups of users. Rather often, semantic inconsistencies also appear even within one single molecular biology database.

Solutions? Many researchers (cf. [63], [7], or [71]) suggest, that an ontology may help to conquer at least the semantic inconsistency of biological terms (semantic heterogeneity). [63] describes an ontology as "a concise and unambiguous description of what principal entities are relevant in an application domain and how they can relate to each other". Thus, an ontology is more intended to be a collection of controlled vocabulary. This would ensure e.g. a semantically consistent access to semantically consistent data.

4.1.3 Updates

Molecular biology database integration systems need a regular (approximately daily) update. Biologists, of course, want and need to access the most up-to-date data.

ASCII flat files are the standard data exchange format. Most databases offer their entire data for download via the web. The differently structured flat files have to be transformed into the integrated system's format. Of course, this should be done automatically by parsing and transformation tools. But as [67] stated, a fully automated transformation is hard to implement, because of the following reasons:

- translation procedures must cope with inconsistencies on both the syntactic and semantic levels within one database
- schemas and flat file formats alter frequently – sometimes unannounced or undetected

- schema heterogeneities of the data sources may require e.g. to follow transitive relationships within one or several databases to collect all data for a particular object – this is complex to implement and therefore time intensive and error-prone

- implementation of parsing tools is time intensive and error-prone

Solutions for these problems have to be discovered or completely new ways of data acquisition have to be developed. But so far, beside flat files, there is no alternative molecular biology data exchange format in sight.

4.1.4 Semistructured Data

Another major problem that makes molecular biology database integration a non-trivial task is a property of the data. As mentioned above, (almost) all molecular biology databases in their entirety are available as (ASCII) text files. Therefore ASCII is the *de facto* standard of data exchange for molecular biology databases.

Some of these data are completely unstructured (like images, raw text), others are well structured, and the rest is in between. These "in between" data are almost structured but may be imperfect (i.e. incomplete), irregular, redundant, or may contain errors. Most of them are implicitly structured, i.e. they are not explicitly but due to their meaning intrinsically structured. In addition, the data may evolve frequently. Therefore molecular biology data are good candidates for the semistructured data modelling approach [2].

Systems for molecular biology database integration should be able to model these semistructured data. Usually, data models of common DBMSs are not adequate for this task. ACEDB (cf. Chapter 3.3.2) is more suited to model semistructured data, but this should be no wonder as ACEDB was developed especially for molecular biology data.

As it has sometimes been noticed, the eXtensible Markup Language (XML) is likely to be a useful approach to model biological data, especially in the context of molecular biology database integration. XML is well suited to model semistructured data and data stored in text files. Several other researchers have already begun to think about and investigate the use of XML in modelling biological data, especially in the context of molecular biology database integration (see below).

4.2 Classifying the Current Work on Database Integration

In general, when speaking of database integration, one can distinguish between two common ways of integration:

- A *federated* approach does not integrate the databases physically nor does it impose a common schema on the member database. The data continues to reside in separate databases but a software layer makes them act as a single integrated data collection [46].

- A *warehousing* approach physically integrates the data from various sources into a single data source.

Note: this classification is according to the integration strategy and should not be mixed up with the concepts of a *federated database system* or a *data warehouse* common in the computer science database community.

Molecular biology database integration approaches mostly choose the federated approach although the performance is less high compared to the warehousing approach. The main argument for using the federated approach is the update problem (see above). Warehousing approaches are more difficult to keep up-to-date.

In the following we give a more detailed classification of molecular biology database integration approaches, focusing on its implementation. According to [39], molecular biology database integration approaches can be classified into several classes. We extend this classification by the class "Theoretical Approaches". Under this name we understand approaches that try to develop ontologies or standard exchange formats. These approaches are mentioned here, for the sake of completeness and because of the proposed importance of controlled vocabularies to capture the problem of semantic inconsistency. A nice review of candidates for data exchange formats in molecular biology is [51]. These theoretical approaches usually are not yet running systems that integrate distributed databases. But they could guide the way to such a desired infrastructure. E.g. the TAMBIS project (cf. Section 4.3.8) is a promising example for a molecular biology database integration system that is based on an ontology. Maybe other existing ontologies (many of them are still under construction) are eventually extended to integrated systems.

The classification is described in the following. We use the term "member database" for a database which is integrated in a running system.

Theoretical Approaches

As mentioned above, there are approaches to develop standardized exchange formats and ontologies summed up in this class.

Even in the early years of molecular biology databases, single research groups attempted to standardize the format of the databases (e.g. [28] or [5]). Current approaches are based on XML to address the semistructured feature of biological data (see e.g. [72]).

The usefulness of ontologies are already discussed above. It should be mentioned, that an underlying ontology is also very helpful for a single molecular biology database. A problem regarding ontologies may be posed by the fact that the definition of a controlled vocabulary – typically done by a group of experts – is probably critically reviewed or even refused by other experts.

Examples: Ontology for Molecular Biology (OMB) ([63]), Immunogenetics Ontology (IMGT-ONTOLOGY), XML based standardization approaches such as: GXML (cf. Section 7), BSML (cf. Section 4.3.9), StarDOM (cf. Section 7).

Navigation Through Hypertext Links

This approach provides links between an entry of one member database to an entry of another member database. Usually it follows the federated approach of integration. Users can interactively navigate from one entry to the other by traversing these links. Usually, users can only search within one database to find a starting entry. They can then request to see all linked entries from another database.

The hypertext links currently used in molecular biology databases are, according to the XML-XLink [73] terminology, simple outbound links. More sophisticated hypertext links, and linkbases as defined in XLink, would significantly improve the "linkage" of molecular biology databases.

Links are generally provided by each member database. In some cases the links are added by the navigation system. This approach usually yields an information retrieval system which provides fast indexed access to flat file databases, rather than employing a DBMS.

Examples: DBGET/LinkDB (cf. Section 4.3.2), Entrez (cf. Section 4.3.4), Sequence Retrieval System (SRS) (cf. Section 4.3.7).

Data Warehouses

In a data warehouse a set of heterogeneous databases are translated and physically loaded into a single database (the data warehouse). For each integrated database a translation from its format and conceptualization into the format and conceptualization of the warehouse must be defined. Thus, the conceptualization of the warehouse database subsumes the conceptualizations of all member databases.

The data are stored locally in the warehouse database following the warehousing integration approach. This makes query processing very fast. On the other hand, this approach must overcome the update problems (see above). In fact, few research institutes have organized their data from different projects and research teams (within the same research institute) in a data-warehouse-like architecture (e.g. MIPS, cf. Appendix A.63).

Example: Integrated Genome Database (IGD) (cf. Section 4.3.5).

Multi-Database Querying Systems

Multi-database querying systems usually are federated integration approaches. Users are allowed to construct complex queries and to perform them against multiple, physically distinct, heterogeneous databases. A query must explicitly identify both the member database that it uses, and the database components (e.g. tables and attributes in case of a relational DBMS) that are queried within each database. Queries can include requests on several databases. This kind of query interface is of course not very transparent.

Example: BioKleisli (cf. Section 4.3.1).

Federated Database Systems

The term "federated database system" has a number of connotations. See [19] for a comprehensive discussion. It implements the federated integration concept. In general the term denotes a collection of autonomous database systems which cooperate with each other.

The federated approach does not force member databases to be physically integrated within a central DBMS. It rather defines interfaces between a central federated schema and the schemas of the member databases. Queries are translated from the federated schema to the schemas of the member databases (in case of relational DBMSs e.g. using views) (cf. [19] for details about federated database systems).

The approach of federated database systems is a widespread approach within the computer science community. Several application domains use the federated approach to integrate different data sources and associated tools. But the bioinformatics community paid only little attention to it. In fact, currently there is no system for molecular biology database integration is implemented as a federated database system. There are several reasons for this fact, including (among others):

- Molecular biology database schemas undergo continuous evolutions. Schema evolutions in federated database systems are problems that are currently investigated. But so far there is no mature approach to adequately deal with schema evolutions.
- Federated database systems integrate only relational databases. But most database providers offer their entire data only in flat files.
- A federated database system is complex and costly to implement.
- The complexity of molecular biology databases and the poor documentation is a great handicap for computer scientists to start research projects in this domain.

The only molecular biology database integration approaches that adopt at least some concepts of the federated approach is TAMBIS (cf. Chapter 4.3.8) and the OPM multi-database tools (cf. Chapter 4.3.6).

4.3 A Selection of Current Work on Database Integration

We present a short description of current molecular biology database approaches. The selection is randomly and may represent the most important and best known approaches. The list includes running systems as well as theoretically approaches and ontologies. The approaches appear in alphabetical order. Note once again, that this list is incomplete. There are several other approaches in this research domain not mentioned here.

4.3.1 BioKleisli

BioKleisli is a system for integrating data from different sources in different formats developed for molecular biology. It is based on a query language called *Collection Programming Language* (CPL). CPL is implemented considering optimization in a distributed environment. The resulting system is called CPL-Kleisli. BioKleisli is an instantiation of CPL-Kleisli, which is used to support the Human Genome Project at Philadelphia Center for Chromosome 22 [20].

BioKleisli is available at the University of Pennsylvania: `http://agave.humgen.upenn.edu`

Data representation. Data in BioKleisli are represented using the Collection Programming Language (CPL), developed for this system. CPL is based on the idea of type orthogonality. Thus its design is structured around its type system. CPL is not object-oriented. Basic types of CPL include the common types such as boolean, integer, string, etc. CPL provides set, bag (multi-set), and list as collection types. In addition, the CPL type system allows arbitrary nesting of collection types together with records and variant types. The type is given by the syntax [20]:

$$\tau := \texttt{bool} \mid \texttt{int} \mid \texttt{string} \mid \ldots \; |\{\tau\}|\{|\tau|\}|\{\|\tau\|\}| < l_1 : \tau_1, \ldots, l_n : \tau_n > |[l_1 : \tau_1, \ldots, l_n : \tau_n]$$

were

- $\{\tau\}$, $\{|\tau|\}$, and $\{\|\tau\|\}$ respectively construct set, bag, and list types from the type τ,
- $< l_1 : \tau_1, \ldots, l_n : \tau_n >$ constructs variant (exclusive) types from types $\tau_1 \ldots \tau_n$, and
- $[l_1 : \tau_1, \ldots, l_n : \tau_n]$ constructs record types collecting objects of types $\tau_1 \ldots \tau_n$.

For each type there are two kinds of primitives: constructors and primitives for decomposition (called "destructors"). The syntax for constructing values of a type is:

- $\{v_1, \ldots, v_n\}$ for sets (similar for bags and lists),
- $< l = v >$ for variants, and
- $[l_1 = v_1, \ldots, l_n = v_n]$ for records.

Example: A type `Publications` might be defined as in Figure 4.1. An exemplary fragment of data conforming to the `Publication` type is shown in Figure 4.2.

```
Publications = {[title: string,
                 authors: {|[name: string,initial: string]|},
                 journal: <uncontrolled: string,
                           controlled: <medline-jta: string,
                                        iso-jta: string,
                                        journal-title: string,
                                        issn: string>>
                 volume: string,
                 issue: string,
                 year: int,
                 pages: string,
                 abstract: string,
                 keywd: {string}]}
```

FIGURE 4.1: Example type `Publications` in CPL notation (taken from [20])

The type system of CPL (which is slightly richer than described here) is able to express most common data formats including those that contain object identities and arrays [20].

```
{[title="Structure of the human perforin gene",
authors={||[name="Lichtenheld",initial="MG"], [name="Podak",initial="ER"]||},
journal=<controlled=<medline-jta="J Immunol">>,
volume="143",
issue="12",
year="1989",
pages="4267-4274",
abstract="We have cloned the human perforin (P1) gene ...",
keywd={"Amino Acid Sequence", "Base Sequence", "Exons", "Genes,
Structural"}] ...}
```

FIGURE 4.2: A fragment of data conforming to the `Publication` type as defined in Figure 4.1 (according to [20])

The simplest primitive for decomposition are records which use field selection, e.g. p.authors. Decomposition of variants is incorporated in a user-friendly syntax by pattern matching (see [20] for details). For collection types so called monad operations are used. These monad operations are the basis of CPL's implementation language NRC (Nested Relational Calculus) (see [10] and [12] for details on monad operations and NRC).

An example of a simple CPL query is given according to [20]:

```
[title = p.title, authors= p.authors] | \p ← DB
```

The query extracts the title and authors from a database DB. Note the use of \p to introduce the variable p. It is bound to each element of the set DB.

Note, that CPL supports first-class and even higher-order functions. CPL can additionally be extended with a reference type to query databases with object identity (ACEDB, ooDBMS).

System architecture. CPL is implemented on top of a query system called *Kleisli*, which is written in SML [54]. Kleisli is extensible in the sense that it does not depend on CPL. The CPL module of Kleisli can be replaced by a module that supports another high-level query language. The system with the CPL-module is called CPL-Kleisli.

The CPL-Kleisli system consists of two main components, as shown in Figure 4.3 by the dotted line.

The first component provides high-level language support and consists of:

- the CPL-Module: provides parser for CPL programs and translates them into NRC (It is imaginable to replace this module by a SQL- or OQL-module);
- the Type Module: contains routines for type unification as well as inference and implements a parametric type system that supports record polymorphism;
- the NRC-Module: provides expression manipulation routines, which are heavily utilized by the optimizer module, as well as routines for compiling NRC expressions into program calls to the Complex Object Library (see below);

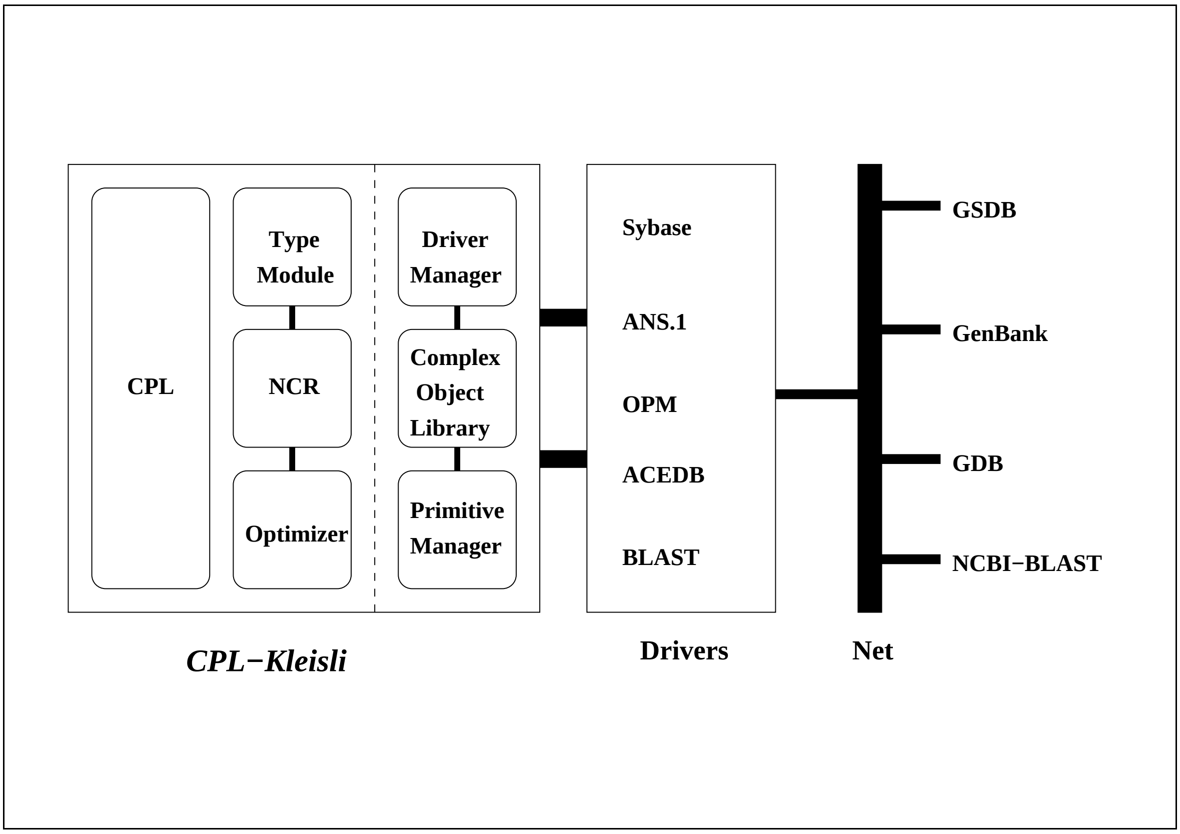


FIGURE 4.3: Architecture of the BioKleisli system (taken from [20])

- the Optimizer: consists of an extensible number of phases, each of which consists of a rule base and a rule application strategy.

The second component is the query engine and consists of:

- the Data Driver Manager: provides an environment for managing data drivers for external data sources.
- the Primitive Manager: provides an environment for managing primitives defined within CPL (macros) as well as primitives imported from external systems;
- the Complex Object Library: provides routines for the manipulation of sets, bags, lists, records, and variants, and forms the core of the query execution engine.

A query submitted to CPL-Kleisli is first processed by the CPL-Module, which translates it into an equivalent NRC expression. The NRC expression is analyzed by the Type Module to infer the most general type for the expression and is then passed to the Optimizer module. Optimized, the NRC expression is then compiled by the NRC Module into calls to the Complex Object Library. The query is executed, accessing data drivers and external primitives as needed.

Capabilities of data retrieval. BioKleisli is a instantiation of CPL-Kleisli, developed in 1994 for the Human Genome Project community. It can be accessed via:

1. Fixed-form query interface: a number of multi-database views are provided through a graphical interface. These views are stereotyped parametrised queries which fire off CPL queries using the input parameters specified by the user.

2. *Ad hoc* query interface: CPL is a query language that provides a syntax closely related to the relational calculus. The developers of CPL call it compact and complete especially for working with deeply nested complex data types.

Further information about the fixed-form query interfaces of BioKleisli can be found in [20]. A graphical user interface set called BioWidgets additionally provides tools to visualize the results of queries (e.g. a genetic map together with annotated sequences can be browsed by the BioWidgets tools).

Discussion. Data representation in CPL-/BioKleisli is closely related to the semi-structured approach of data modelling with XML.

Querying BioKleisli with ad-hoc CPL-queries is not transparent because the user must explicitly identify the databases which should be used and the database components (e.g. tables and attributes in case of a relational DBMS) which are queried.

4.3.2 DBGET/LinkDB

DBGET/LinkDB is an integrated database retrieval system provided by the Japanese GenomeNet service. It is used as the web interface of the databases at GenomeNet, such as e.g. KEGG, accessible at: `http://www.genome.ad.jp`

Fundamentals. [26] states that common computer science approaches in database integration – such as converting data from different sources into a unified schema – have not been practical to cope with a large number of databases with continuous format changes. As molecular biology databases undergo continuous data format changes [26] suggests to integrate molecular biology databases at the level of entries, rather than the level of fields that form an entry. This approach is implemented in DBGET/LinkDB. It is supported by the efforts of the molecular biology database providers: They explicitly enter cross-reference information of related entries in other databases into their own database.

The DBGET/LinkDB system links different molecular biology databases. The concept of a link in this context is very important. There are two general classes of links in DBGET/LinkDB:

- links provided by the original databases (also called direct links);
- computed links, subdivided into:
  - neighbors and similar entries: computed by similarity search programs;
  - indirect and reverse links: computed by using the direct links; (an indirect link is a link between two entries that are linked via at least two direct links; a reverse link is a link reverse to a direct link)

DBGET/LinkDB intends to compute and provide both kinds of links. Currently, not all similarity links are computed and integrated with direct, indirect and reverse links.

[26] describes the characteristics of the system as follows:

1. distributed databases are integrated with a transparent interface;
2. manipulation of flat file databases is done at the level of entries;
3. other (e.g. local) databases can be integrated, links to other databases (outside DBGET/LinkDB) are provided;
4. two different interfaces are provided (a web interface for fixed-form queries and a UNIX-like interface for free-form queries);

System architecture and data representation. The DBGET/LinkDB system consists of three packages:

- DBGETclient (also called NetDBget) provides the UNIX interface;
- DBGETserver maintains the databases locally;
- WebDBGETserver provides the web interface;

DBGET/LinkDB has developed an interesting approach of data representation: In general, a database is treated as a collection of entries. These entries may be stored in a single or multiple flat files. Flat files are defined as text files or multimedia files (such as e.g. GIF files). Each entry of a database (or more abstract: of a file) is identified by a unique identifier:

$$\text{database} : \text{identifier}$$

DBGET/LinkDB retrieves all direct links and derives the computed links. All links between related entries are stored in the LinkDB database. These links are represented as:

$$\text{db1} : \text{id1} \rightarrow \text{db2} : \text{id2}$$

This approach is independent of the structure of each entry. The databases are used as they are, there is no conversion of data formats.

In order to accomplish rapid access and search of entries, some auxiliary files are created. This is done by a program called *sequnew* during each update of the system. The auxiliary files are either flat text files or hashed by the GNU database manager library *gdbm*. A C++ class library was created for parsing a wide variety of database formats.

DBGET/LinkDB is able to integrate both local and network flat file databases. The network configuration of distributed databases is defined in an auxiliary file called *dbtab*. This file contains a table with the following format:

database	dbtype	dbsite
name of the database	format of the database	location of the database

The database format column provides the information which parsing program to use. The location of the database may be the address of another server or a local directory.

DBGET/LinkDB is available as a system for local installation or via the web (server at GenomeNet, see above)

Capabilities of data retrieval. DBGET/LinkDB provides the following query interfaces:

1. Fixed-form query interface: this interface is web-based and provides keyword search against a predetermined database.

2. Free-form query interface: this query interface is provided by local installations. The DBGET/LinkDB can be searched via a UNIX shell using a home-made query language. This query language consists of only five commands (cf. [26]).

There are no capabilities to browse through hierarchies. The entire integrated amount of data cannot be downloaded as flat file(s).

Discussion. DBGET/LinkDB is an integration approach providing links between entries of the member databases. It does not address the problems of heterogeneity. There is no underlying ontology used for keeping the data consistent. Query interfaces are limited to simple fixed-form and keyword queries. There is no powerful *ad-hoc* query language accessible via the web (only available for local installations).

The query interface is also not transparent. The user must identify the relevant source to perform an initial query.

4.3.3 EBI's Approach Using CORBA

The European Bioinformatics Institute[1] (EBI) developed an approach to molecular biology database integration using the *Common Object Request Broker Architecture* (CORBA) [31].

Biologists use several different hardware and software platforms, operating systems, DBMSs, and programming languages. CORBA is a *de facto* standard that offers the opportunity to make such differences transparent and helps to combine heterogeneous data sources and application programs [38]. It is a standard for defining interfaces and invoking methods through them. With the use of CORBA data sources and application programs interact through interfaces which hide the underlying implementation details. For an introduction to CORBA see e.g. [64].

The two main concepts of CORBA are the specification of:

- the *Interface Definition Language* (IDL), which provides a language-independent way of describing public interfaces, and

- the *Object Request Broker* (ORB), which transparently transmits requests from clients to object implementations.

[1] `http://www.ebi.ac.uk`

Once an interface for a distributed object is defined in IDL, any client can use it depending only on that public interface of the object.

To develop the interfaces of objects it is often reasonable to carry out a use-case analysis. The resulting interface of an object can have strong similarities to the conceptual model and also to the database schema of the object but the three data models should not be confused. The conceptual model should model the data as they really are. The IDL interface is independent from implementation details but it represents an application specific view. The database schema is specific to the used DBMS (e.g. relational or object-oriented) [38].

In general, databases currently do not provide CORBA interfaces to their data. As a consequence, a wrapper which implements IDL interfaces is needed. The EBI developed and implemented several CORBA wrappers for databases such as EMBL (cf. Appendix A.28) and RHdb (cf. Appendix A.94) as well as for several applications (analysis tools).

4.3.4 Entrez

Entrez[2] is an integrated database retrieval system for molecular biology databases. It was developed at the National Center for Biotechnology Information (NCBI). Entrez provides access to nucleotide and protein sequences as well as to the literature. It can be accessed at: `http://www.ncbi.nlm.nih.gov/Entrez`
A CD-ROM package containing data and software is also available, issued six times per year.

System architecture and data representation. Entrez is introduced consisting of different worlds, each containing data records. The worlds in Entrez include:

- Nucleotide Sequences (DNA sequences with annotations)
- Protein Sequences (amino acid sequences with annotations)
- Three-Dimensional Structures (protein structures)
- Genomes (chromosomal and mapping information)
- Taxonomy (classification of nucleotide and protein sequences according to their organism)
- PopSet (set of aligned sequences resulting from dedicated studies, representing events such as evolution and population variation)
- PubMed (bibliographic records from MEDLINE)

Each world has a different notion of similarity.

Integration consists of cross-references provided as hypertext links to data within one world and to data in other worlds. After an initial query is completed, the user can follow these cross-references to related information. Cross-references include:

- “Neighbors”: computed links based on the worlds’ notions of similarity

[2] All information about Entrez was acquired from `http://www.ncbi.nlm.nih.gov/Entrez` and [65].

- "Hard Links": links already existing in the member databases

Similarity links are computed e.g. using the BLAST program to find local sequence alignments (in the DNA and Protein world). The entire sequence databases are compared against themselves in this way.

System architecture and data representation. All data from the integrated databases are stored locally. Each data item (entry) is represented in ANS.1 [5] format. Thus, the data from different sources are converted into ANS.1. The integration of all entries in the member databases is done via a so-called Integration Database (ID). Each entry is referred to by its unique identifiers stored in the ID.

For further information see [65].

Capabilities of data retrieval. Entrez provides a fill-in form to perform keyword search or express simple boolean queries (using AND, OR, and NOT). In addition, a more complex fixed-form-query interface is provided where the user can specify the data fields to be queried. Each category of the Entrez world has its own fixed-form query interface.

There is no *ad hoc* query interface.

Discussion. Entrez is an integration approach providing links between entries of the member databases. It does not address the problems of heterogeneity. There is no underlying ontology that keeps the data consistent. Query interfaces are limited to simple fixed-form and keyword queries. There is no powerful ad-hoc query language.

The query interface is also not transparent. Though the user do not need to identify the relevant data sources for a query, the world in which the first query is answered must be identified.

Nevertheless, Entrez is a commonly used system in molecular biology, offering access to a great part of available biological information.

4.3.5 The Integrated Genome Database (IGD)

The Integrated Genomic Database (IGD) is an international project to develop an information management system for human genome researchers which interconnects existing molecular biology databases and analysis tools in a data warehouse. IGD is an international project which now seems to be given up. The URL of IGD which is provided by several survey sites is a dead-end.

The following information about the IGD approach of molecular biology database integration is retrieved from: `http://www.ai.sri.com/pkarp/mimbd/94/abstracts/ritter.txt`
and
`http://www.igd.fhg.de/archive/1995www95/proceedings/posters/18/`

IGD is designed as a network system based on a client/server architecture. With regard to the origin and scope of data, the system can be subdivided into three levels:

- resource databases which contribute data

- target database servers which manage the integrated data

- front-end clients which manage data locally to the user

Data from the IGD resource databases, called Resource End Databases (REDs) are periodically collected, reformatted and exported to several IGD servers, or target databases, called Target End Databases (TEDs). All TEDs have the same conceptual schema, but may differ in the physical implementation.

Two versions of IGD were/are implemented using the relational DBMS Sybase and ACEDB as DBMS. The DBMSs are used in a complementary manner which combines the strengths of each system.

All the IGD servers contain more or less the same data (under weak consistency constraints), and all data are read-only for end users. New data generated at the end user (client) site are forwarded (in appropriate format) to the relevant source databases using the usual submission channels for the database in question.

Users interact with the IGD system through a set of locally installed “Front End” tools. The most important parts of the front-end IGD are the local database manager and the interfaces to communication and analysis. Users can query the IGD servers and download the resulting data into their local database, where it can be manipulated and analyzed. Private data and analysis results may also be deposited into the local database.

Local front-end IGD databases provide read-write access for local users. In their local database, biologists can annotate and edit retrieved public data, and link it with their local experimental data. Thus experimental data can be kept in their public context. The IGD local database is implemented using the ACEDB genome database software, and has identical schema to that of the IGD target databases.

Discussion. The IGD data warehouse is probably not in use any more. Thus, it is difficult to evaluate this approach.

In general, it can be stated, that the data warehouse approach is surprisingly frequent in use to integrate molecular biology data sources within a research institution. Those data warehouses integrate only small parts of the existing biological data (usually only the data which is acquired within the project or research institute) and reflect the primary foci of the projects or research institutes. An example is the MitBASE (cf. A.64) database, which integrates mitochondrial DNA data from different species.

Currently there is no system implemented as a data warehouse, that integrates general, heterogeneous molecular biology sources.

4.3.6 The OPM Retrofitting and Multi-Database Tools

The Object-Protocol Model (OPM) is an extended object model developed for biological database applications. Details about OPM are described in Chapter 3.3.3 and [16].

The authors of OPM also developed the so-called “OPM data management tools” for creation and accessing databases using OPM, constructing OPM views of existing relational databases and structured files, and querying databases through uniform OPM views [48]. These OPM data management tools consist among others of so-called “OPM retrofitting” and “OPM multi-database” tools.

The retrofitting tools can be used for constructing and maintaining OPM views on top of existing flat files or relational databases that were not using OPM. Constructing such views follow an iterative strategy [48]:

1. A canonical (default) OPM view is created automatically from the underlying database schema.
2. This canonical OPM view is refined using schema restructuring operations such as renaming or removing classes and attributes, merging and splitting classes, etc.
3. A mapping dictionary documents the information regarding the relationship between the OPM view constructs and the corresponding representation in the underlying database.

The multi-database tools use the mapping dictionary recording the metadata needed for accessing databases and information needed for following inter-database links.

System architecture. The OPM multi-database tools consist mainly of [48]:

- A java-based *OPM multi-database schema browser* allows browsing the schemas of multiple databases.
- A web-based *multi-database query interface* provides capabilities of data retrieval (see below).
- A *OPM multi-database query processor* interacts with database servers for each member database (see next item). It generates queries over the member databases from the initial queries expressed in the OPM query language (OPM*QL).
- Each *database server* provides DBMS specific query translation functions and manage the execution of single database queries expressed in OPM*QL.

The tools are implemented in an object-oriented environment using primarily C++, Java and CORBA.

Capabilities of data retrieval. Databases implemented using OPM can be queried through an *ad hoc* query interface using the home made query language OPM*QL having a SQL-like syntax. The OPM database query tools also provide an interesting query interface that is a mixture of a fixed-form query interface and a capability to browse through a hierarchy. The user must chose a database and can browse through the OPM view of this database using the OPM schema browser and identifying relevant classes and attributes for his request. Thus, a query tree which leaves are primitive attributes is generated. A fixed-form query form is then created enabling the user to specify query conditions (keywords that are queried against the leaves of the query tree).

Most likely, the OPM data management tools do not provide flat file releases of the underlying databases.

Discussion. The OPM system to integrate heterogeneous databases uses a middleware layer to apply a uniform object model on the member databases. The query capabilities can be transparent in such a way that the user can chose the databases he wants to query but can also express transparent *ad hoc* queries.

The OPM system follows the federated approach. By using (OPM) views to integrate heterogeneous databases it is closely related to the approach of a federated database system.

Note that the new generation OPM database management tools are now commercial products of Gene Logic Inc.[3] [46]. They have been used to construct a prototype called "Molecular Biology Database Federation" [48] that integrates GDB (cf. Appendix A.35), GSDB (cf. Appendix A.38), and GenBank (cf. Appendix A.36).

4.3.7 Sequence Retrieval System (SRS)

The Sequence Retrieval System (SRS) [24] was designed by the European Molecular Biology Laboratory in the early 1990 to support rapid and integrated access to many distributed flat file molecular biology databases. It is now maintained by Lion Bioscience[4] and available as a software package (commercial). There are several SRS servers installed on different web-pages all over the world. EMBL, for example, provides an SRS server. It can be accessed free of charge at: `http://srs.ebi.ac.uk`

System architecture and data representation. SRS uses the metadata of each member database to build searchable indexes for each field in each database and link indexes between all member databases. As a consequence, cross-references between entries within one member database and between entries in different member databases are provided.

New databases can be added to local SRS installations. For each added database, SRS must be told about

- the syntax of the entry and the data fields
- the logical structure of the database, i.e. the name and the type of the database and its fields, as well as how to index field contents and links to other databases, etc.

The structures of the member databases are represented by Backus-Naur Form (BNF) grammars. SRS parses the flat file release of each member database to create the search indexes according to the database schemas. The representation of the member database schemas using BNF grammars simplifies problems with rapidly evolving database schemas. A flat file structure change of one database is carried out by changing the grammar for this database schema [14]. The grammars are internally represented using the so-called Icarus language [36].

Capabilities of data retrieval. The user must determine which data sources are relevant for the query session. SRS offers several different query interfaces:

1. Fixed-form query interfaces:
    - searching a string against all databases
    - choosing an index by browsing the data fields of the determined databases and searching for a string in that field
    - searching terms against four different data fields at once via a standard query form
    - searching at least one of all common fields by a search term via an extended query form

[3] `http:www.genelogic.com`
[4] `http://www.lionbioscience.com`

2. *Ad hoc* query interface:
   SRS can be searched using the home-made SRS query language which is based on the Icarus language [36]..

SRS also provides the capability to generate views [14]. Users can identify search indexes to be hidden in the representation of a query result.

Discussion. SRS is an integration approach providing links between entries of the member databases. It does not address the problems of heterogeneity. There is no underlying ontology used for keeping the data consistent. Query interfaces are limited to simple fixed-form and keyword queries. The SRS query language has not the power to search directly against all member databases. In addition, the use of the query language assumes that the user provide knowledge about the format of the database being queried. This is a complex task because the schemas are not transparent. To utilize the entire power of the query language, the user must know the format of each member database.

As the user does not need to identify the relevant data sources for a query, the query interface is also not transparent. A detailed but very complex user manual provides information on how to take advantage of the full power of the system.

An extension of an existing, local SRS system by adding new data sources seems to be costly. At least there is a (complex) administrators manual.

Nevertheless, SRS is a commonly used system in molecular biology because many flat file databases are most conveniently accessible through SRS.

4.3.8 The TAMBIS Project

TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources) is a project which aims to aid researchers in biological science by providing a single access point for biological information sources round the world. The central module is the so called biological Concept Model, a knowledge base comprising an ontology of bioinformatics terms.

TAMBIS is under development at the University of Manchester and accessible at: `http://img.cs.man.ac.uk/tambis`
A password mechanism limits the load on the local server. An account is free of charge.

Data representation. The ontology is represented using the GRAIL Description Logic [58] that was originally developed for the modelling of medical terminology for a system to support clinical user interfaces. A description logic is an isa-based classification system that allows a recursive, compositional model to be built from terms and binary relations [6].

GRAIL models application domains in terms of concepts (classes), roles (relations), and individuals (objects). Concepts and roles are called terms. Compositional concepts can be built up using recursive term constructors. All roles in GRAIL are bi-directional.

The GRAIL model of TAMBIS consists of three parts:

- Assertions: the model contains a collection of elementary concepts which cannot be decomposed further.

- Operations and Reasoning Services: GRAIL provides a collection of operations which allow the construction of compositional concepts and roles. Each concept is classified to determine the precise hierarchical position.

- Sanctions: GRAIL provides rules (called sanctions) to restrict the construction of complex concepts to only those that are semantically meaningful. These rules dictate which roles may legitimately be applied to which concepts. A composition is not allowed unless it is explicitly sanctioned. But sanctions represent the possibility of composition, not its necessity.

GRAIL supports multiple inheritance. Thus, classification in the description logic is different from traditional classification in a tree-like structure. GRAIL is more flexible (concepts can be classified with many parents) and as a consequence, it supports multiple views of the same concept [7]. Sanctions are also inherited.

By combining existing concepts to create compositional concepts is a very powerful and flexible approach. A large number of concepts can be created from a relatively sparsely populated model. A GRAIL model is not a static tree, but should be considered as a resource that can be queried by applications [7].

In TAMBIS the biological Concept Model is used to (cf. [6]):

- describe the metadata of the underlying data sources, representing an over-arching universal schema

- express queries in the modelling language

- drive a GUI for query formulation

- mediate between the various data sources by exploiting the biological concept hierarchy (. . .)

System architecture. TAMBIS is designed following the three layer model of a classical mediator/wrapper architecture (cf. Figure 4.4).

Layer 1 consists of the Concept Model – i.e. the knowledge base of biological terminology – and a knowledge driven user interface. The user can combine terms from the Concept Model to form declarative, source independent queries.

Layer 2 is a mediator that identifies the eligible sources (Source Model) and rewrites the query to a series of source dependent ordered procedures (query transformation module).

Layer 3 consists of external sources wrapped with a consistent structural model (query execution).

Capabilities of data retrieval. TAMBIS provides a completely different approach of data retrieval compared to most other molecular biology databases and database integration systems. There are no *ad hoc* or fixed-form query interfaces. At first sight, data retrieval looks like indirect browsing a classification. But it is rather different.

TAMBIS provides a knowledge-driven, form-based graphical user interface (GUI). Via the interface, the terminological model can be viewed, and the user is guided in query formulation and manipulation.

A query is formulated and manipulated as follows: The user browses the model to identify a concept of interest based on which a query should be formed. A Query Manipulation tool provides the user

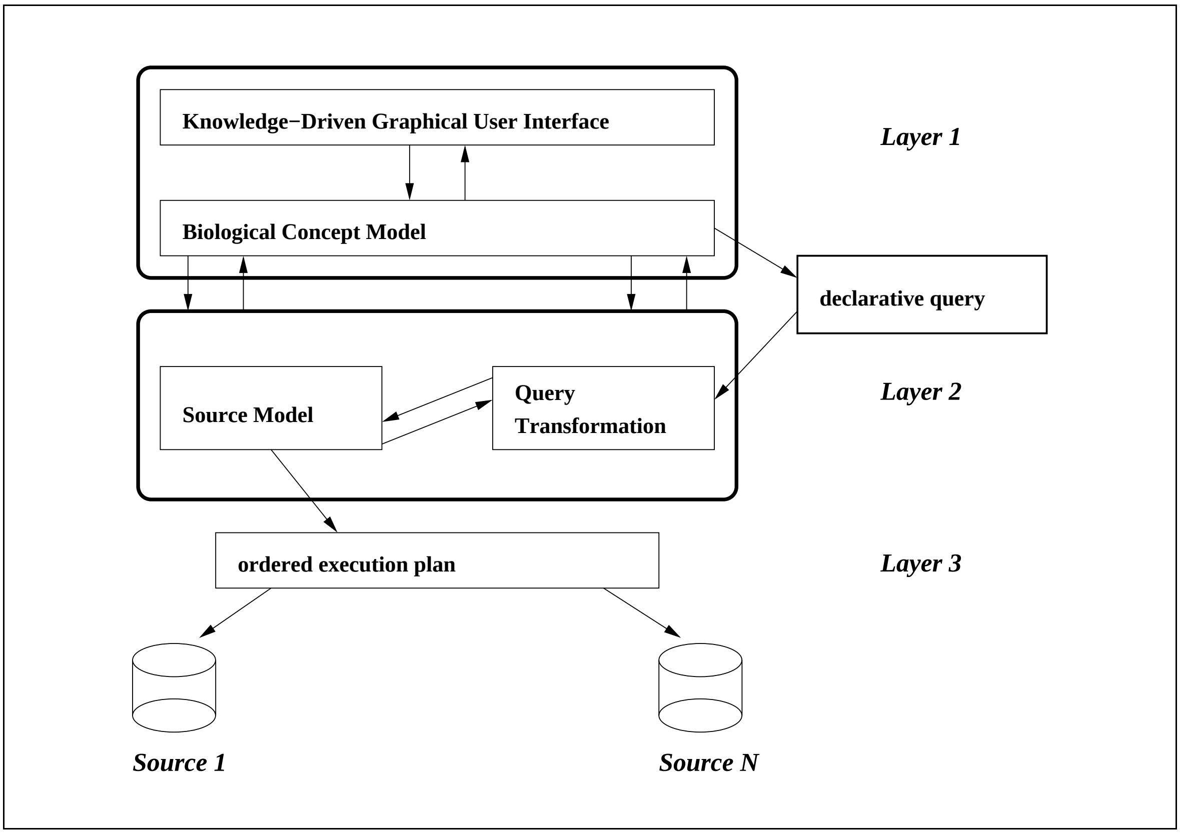


FIGURE 4.4: TAMBIS three layer, mediator/wrapper architecture [6]

with more information about the concept, especially all legitimate criteria that can be applied to the concept. (This is also called specializing the concept.) The user may choose one or more of those criteria to construct the query. E.g. the concept motif may be specialized by the criteria isComponentOfProtein. The query motif isComponentOfProtein corresponds to the English expression "find all protein motifs" [6].

A query can be manipulated by adding or removing criteria, or by replacing terms with more specialized or more general terms. More details about the TAMBIS query interface and query processing can be found in [55]

Note that in TAMBIS the term concept is interchangeable with the term query. Constructing a concept within the model (i.e. a description of what the user wants) means constructing a query ("find all things that fit to the description").

In GRAIL queries are expressed declaratively and source independent. Layer 2 of the TAMBIS system (see above) plans query execution and translates a GRAIL query into a CPL (Collection Programming Language as described in Section 4.3.1) program. Details about query translation can be seen in [6].

Discussion. TAMBIS seems to be the most promising approach of molecular biology database integration. The biological ontology ensures both

- a consistent integration of different schemas without forcing a global schema and

- a consistent query formulation process.

The user interface of TAMBIS is very transparent. The user is freed from managing and executing all tasks listed above in the context of transparency.

A great credit of TAMBIS is the query interface. Common query interfaces have a big disadvantage: *Ad-hoc* query interfaces are powerful but often too complex for biologists, because they usually do not provide profound expertise in querying a database using a query language. Thus, most users do not know how to express queries or cannot make use of the full power of the query language. Fixed-form query interfaces on the other hand are (usually) simple but in return less powerful. Users may wish to have a special view on the database(s) which is not provided by the interface.

The TAMBIS interface combines the advantages of the two classical alternatives. It is simple in the sense that the user do not need to provide knowledge about a special query language and its theoretical background. And on the other hand, it is very powerful, because the user can create own (integrated and transparent) views on the databases and can even extend existing views by extending the model.

4.3.9 XML Based Approaches

BSML

The Bioinformatic Sequence Markup Language (BSML)[5] and the BSML Browser and Tools are XML applications for management, communication, and interactive visualization of genomic data. These applications are developed by Visual Genomics, Inc. (now seemingly renamed to LabBook, Inc) under a grant from the US National Human Genome Research Institute in 1997. Information about BSML presented here was gained from the LabBook web-site at: `http://www.labbook.com`

BSML is an extensible language specification and container for bioinformatic data defined using on XML. It is intended to serve as a public domain standard in modelling biological (especially genomic) data. Project data and links can be stored in BSML "documents". Following the XML philosophy these "documents" are composed of two distinct sections:

- The **Definition Section** encodes the bioinformatic data.
- The (optional) **Display Section** encodes information for the graphical representation of the data.

This has several advantages:

- Graphical representation of a BSML document is independent of BSML creation and storage.
- Data are rendered graphically in the BSML Browser providing a dynamic and interactive user interface.
- Genomic data can be visualized and underlying, linked data sources can be accessed easily.

In addition, this approach supports consistent multi-user access and easy data sharing: BSML documents can be viewed and transmitted via networks and internet using standard protocols.

[5] LabBook Inc., `http://www.labbook.com`

The BSML Tools of LabBook provide data analysis tools which create new and persistent BSML output. Some of these tools are commercial, basic applications like a prototype of the BSML Browser are for free.

Based on BSML LabBook intends to develop an inter-operating bioinformatics infrastructure providing access to several molecular biology data sources. The BSML language specification can be seen and evaluated by registered users at the LabBook web-site under: `http://www.labbook.com/products/standards.asp`

GXML

The Genome-oriented eXtensible Markup Language (GXML) [67] is a markup language, defined in XML, that can be used to model and exchange genomic data. GXML was developed at Osaka University to model the special properties of genomic data (see [67] for details).

The GXML DTD (see [67]) defines several major elements for markup. These major elements include:

- genome: is the root element for a single genome
- contig: contains the nucleotide sequence for a contiguous region of the genome
- feature: contains information on a gene, such as gene name, location, DNA sequence, amino acid sequence for the encoded protein, and function
- pw: contains information on a specific pathway
- role: contains information about a specific enzyme

To optimize extracting and processing genomic data contained in GXML documents, [67] introduced a new query language called Genome-oriented Query Language (GQL), based on XML-QL. A typical GQL query consists of a `WHERE` clause and a `CONSTRUCT` clause. The first clause specifies the structural elements which are to be extracted or processed. The second clause specifies how the results will be structured.

In addition, GQL provides an extended genomic view of the underlying data (see [67]). This view allows users to form queries using biological meaningful constructs without the need to care about the implementation of the data in the underlying GXML document.

A prototype system based on GXML and GQL was implemented [67]. A Perl-script extracts data from GenBank (cf. A.36) and KEGG (cf. A.52) databases and converts them into GXML documents. A GQL query processor was implemented in C++.

This prototype shows promising test results [67]. Future work will include query optimization (especially in executing joins), handling update problems, the investigation of the use of XLink and XPointer to allow more efficient representation of relationships between GXML documents, and the extension of the GQL functionality by the capability to calculate transitive closures [67].

StarDOM

Several molecular biology databases (e.g. PDB, cf. A.75; BMRB, cf. A.10) use the “Self-Defining Text Archive and Retrieval” (STAR) format [32] to store structural, christallographic diffraction, and NMR data. Most likely, the STAR format is a standard data format for structural biology data.

StarDOM is a software package for conversion of STAR files into XML files developed by the European Molecular Biology Laboratory (EMBL) in Heidelberg/Germany [37]. This allows interactive navigation of the data by dedicated browsers, as well as easy querying by XML query languages. These are preferred ways of accessing biological data stored in databases.

The transformation of STAR files into XML is performed as follows [37]:

- the tree structure inherent in the STAR files is unrolled into an object tree structure
- this tree is described as a Document Object Model (DOM [22]) which describes a tree representation of XML documents
- this DOM is rendered as XML

StarDOM offers a tool for transforming entire databases, which offer their data in STAR files, into XML documents. [37] converted the entire BMRB data set into XML. Using dedicated software these data can be browsed or parsed. Using an XML query language (e.g. XML-QL), complex (and especially declarative) ad-hoc queries can be performed against these data.

Discussion

XML seems to be promising in providing constructs for modelling biological data adequately. Biological data currently are mostly represented in formated files and often appear as semistructured. XML is suited to deal with those challenges. Compared to other formats proposed as exchange standard for biological data such as e.g. ANS.1 [5] (used as data exchange format by Entrez), XML seems to be the better candidate due to the following reasons. ANS.1 requires strict definitions for all aspects of the schema. It is difficult to modify the structure of ANS.1 documents without adversely affecting the operation of tools that operate on them [67]. The ability to easily define, modify and parse the structure and content of an XML document are great merits.

Another advantage of XML over other formats is that XML documents are self-descriptive, i.e. information about the structure of a well-formed XML document can be determined during the parse process [67]. As [67] stated, “this is highly useful for *ad hoc* representations of temporary data, as might be transferred between genome projects”. Consistency between documents is ensured with the use of a Document Type Definition (DTD), which defines a set of rules that govern the structure of the documents.

In addition, XML applications provide powerful data retrieval capabilities. For example XML-QL and XQuery [74] are powerful declarative query languages. By developing dedicated browsers, XML documents can dynamically and interactively be browsed. XML was designed for the web. This makes it predestinated to be the basis for a standard exchange format for molecular biology.

Several approaches picked-up that thoughts. They developed XML DTDs which define modelling languages for different kinds of biological data. Dedicated tools for browsing and querying that data are also provided.

Difficulties may arise from the problem to integrate all current approaches in one agreed, XML based modelling language. But from our point of view, XML is one of the most promising construct to accurately address the problems of molecular biology database integration.

4.4 Summary and Outlook

We described here a list of molecular biology database integration approaches. These approaches were selected to appear in order to cover the whole spectrum of the classification given above. The list probably includes also the most popular systems.

There are still great challenges to meet and high hurdles to overcome in integrating molecular biology databases. We described major problems and requirements in this context. Current systems can be estimated as not mellow yet although there are some interesting approaches.

The semantic heterogeneity of the different data sources is the main problem. Only TAMBIS seems to deal with this aspect. OPM or BioKleisli both using a middle-layer based on a common model (OPM or CPL) do as far as we know not deal with semantic heterogeneity. For example OPM renames classes that have equal terms. This may be desired if both classes are semantically different but otherwise undesired.

The power of the underlying data model varies between the approaches. OPM provides an expressive object model but BioKleisli's CPL modelling language is more powerful. Unfortunately we cannot compare the modelling power of TAMBIS data representation because of lack of information.

The query capabilities of the different approaches are also very heterogeneous. The probably most user-friendly interfaces are provided by TAMBIS and the OPM system.

Performance evaluations of the different approaches are not provided here. Nevertheless, we refer to basic thoughts that a warehousing approach will usually answer queries faster than the federated approach. We want to point out that the performance of the updates should also not be neglected. In the context of updates a federated approach will usually perform better than a warehouse.

In our opinion, an underlying ontology would greatly support to capture the semantic inconsistencies within molecular biology databases. XML seems to be a promising approach for adequate modelling of biological data, because it supports the semistructured data modelling approach.

In addition, we propose that an integrated database system for molecular biology should also provide dedicated analysis tools. An integration of molecular biology databases and molecular biology data analysis tools is desired. A current approach that integrates various molecular biology databases with a large number of analysis tools for biological data is the so called "Biology Workbench"[6] [70]. The system uses the Sequence Retrieval System (SRS, see Section 4.3.7) to integrate the data sources. A collection of CGI scripts perform parsing and converting the input and output of the analysis tools. A second approach to integrate databases and associated analysis tools which is described above is defining common interfaces using CORBA as it is done at the EBI (cf. Section 4.3.3).

A problem which was not mentioned yet, is which databases should be integrated, and which databases should not be considered. Although it is not clear, it is most likely not desirable nor reasonable to integrate all existing (or as much as possible) molecular biology databases.

But still one of the main problems is that the majority of computer science experts are not well aware of this application domain.

The molecular biology database integration approaches described above are summarized in Table 4.1 on page 83. Only the running systems are compared. The approach of EBI using CORBA is not evaluated here. The summery focuses on the major requirements mentioned above.

[6] The Biology Workbench can be accessed via the web at: `http://workbench.sdsc.edu/`

TABLE 4.1 Comparison of the molecular biology database integration systems described above.

Name	Approach	Underlying Ontology	Complex Declarative Queries	Update Frequency	Transparency
BioKleisli	federated; multi-database querying system	no	supported	like member databases	no
DBGET/ LinkDB	warehousing; navigation through hypertext links	no	partially supported	daily	no
Entrez	warehousing; navigation through hypertext links	no	no	daily	no
Integrated Genome Database	data warehouse	no	supported	**??**	**??**
OPM Tools	federated; mediated multi-database querying system	no	supported	**??**	partly
SRS	warehousing; navigation through hypertext links	no	partially supported	depending on the provider	no
TAMBIS	federated; mediated multi-database querying system	yes	supported	**??**	yes

Chapter 5

An Emerging Issue: Modelling of Biochemical Pathways

Molecular biology databases manage primarily data on static aspects of biological entities. Information about single instances of these biological entities is available to analyse and predict general interactions between the entities.

As mentioned previously, molecular biology is interested especially in these interactions. But it is often difficult if not impossible to evaluate and analyse the huge amount of biological data to investigate and predict interactions.

A prominent example is the investigation of genetic regulatory systems. Several genome sequencing projects are leading to nearly complete characterisations of the genomes of diverse organisms. The expression of genes is not a spontaneous action but is regulated by a complex system of several genes interacting with each others. It is usually impossible to analyse the tremendous amount of sequence and expression data to understand these complex systems of gene regulation (also called "genetic regulatory systems" or "regulatory pathways") intuitively. A full understanding requires understanding the quantitative dynamic behaviour of the metabolism.

Computer-based simulation, computation, and prediction would be and already are great benefits to the investigations of genetic regulatory systems and metabolic pathways.

In this Chapter we provide an overview over current approaches in modelling gene expression (regulatory pathways) and metabolic pathways. Both – closely related to each other – are emerging issues of practical relevance because they play a key role in drug design and gene therapy.

5.1 Biochemical Pathways: What They are All About

One special research topic of molecular biology is the genetic information flow. Each organism stores the information important for growth, evolution, and reproduction in genes on the DNA. The genes can be understood as computer programs: the information is kept as linear sequences over a four letter alphabet. Using the genes as blueprints, dedicated procedures called Transcription and Translation can assemble proteins (cf. Chapter 2 for details). The proteins play a key role in growth, evolution and reproduction of any organism, since they supervise and carry out all biochemical reactions.

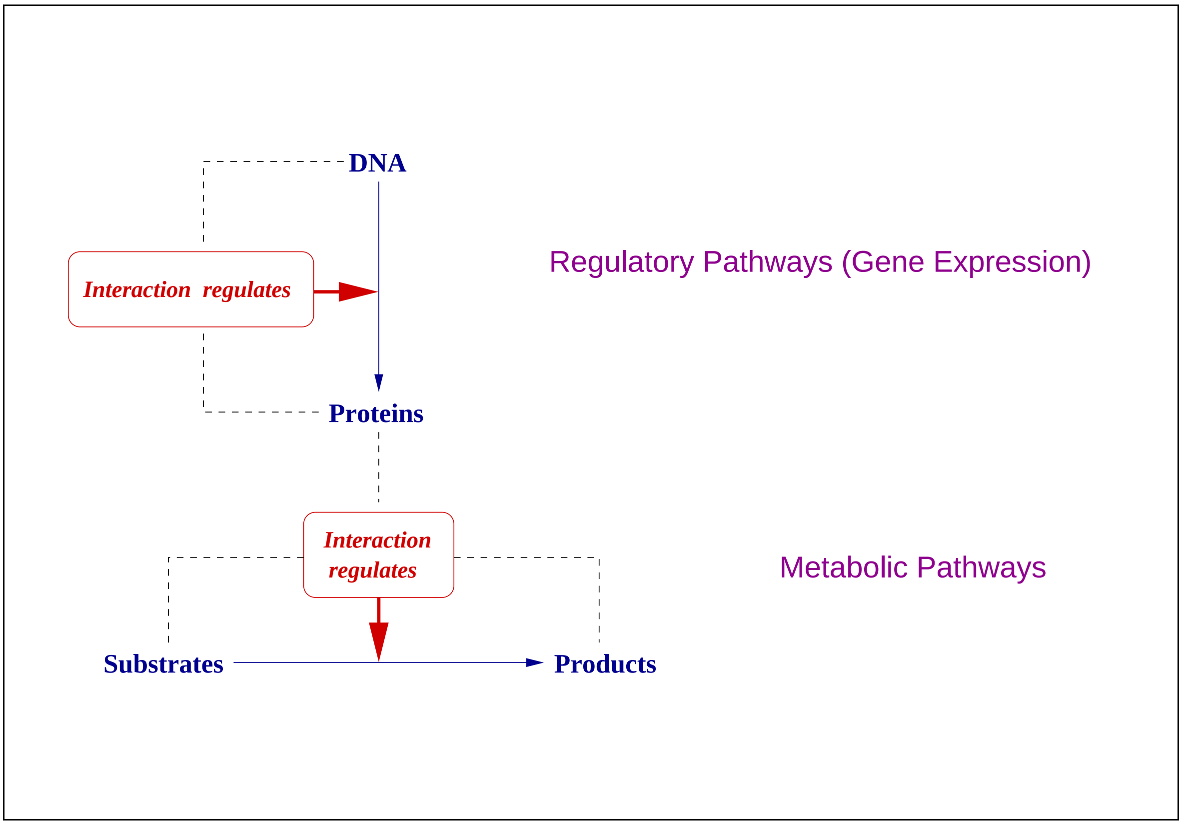


FIGURE 5.1: Regulatory and Metabolic Pathways model the following interactions: Protein-Gene (regulation of gene expression), Protein-Substrates, and Protein-Products (metabolism).

In general, organisms perform thousands of such biochemical reactions. Most biochemical reactions are successive, i.e. they consist of many complex "pathways", that are interconnected with each other. Usually, such pathways represent the successive reactions that transform small molecules into macro molecules (anabolism) or *vice versa* (catabolism). The sum of all anabolic and catabolic reactions is called the metabolism. The reaction pathways within one cell are also called "metabolic pathways". As already mentioned above, the proteins direct and control the metabolism.

An important research objective in molecular biology is the regulation of the genetic information flow and the study (including quantitative simulation, representation, and regulation) of the metabolism. The regulation of gene expression is mainly based on interactions between genes and proteins and reactions of genes and proteins. These interactions and reactions are often called regulatory pathways. The regulation of the metabolism is mainly based on the regulation of gene expression because this determines whether a protein is present to carry out its particular metabolic reaction and reaction-specific kinetics.

Regulatory and metabolic pathways are thus very closely related (see also Figure 5.1) but differ in their properties and particularities. In this section we will describe the particularities of regulatory pathways (also referred to as genetic regulatory networks, Section 5.1.1) and metabolic pathways (Section 5.1.2) from the biologist's point of view. Related to these issues is the investigation of cell communication (in eukaryotes) and other (often more organism-specific) pathways like pathways of mitochondrial protein import or the coagulation of blood. For all these issues, the term "biochemical pathways" is often used. Instead we use this term as the superordinate concept of regulatory and metabolic pathways.

5.1.1 Regulatory Pathways: Regulation of Gene Expression

A representative prokaryotic genome consists of about 4 000 genes whereas the number of genes in the human genome is estimated to be around 100 000 [45]. Usually, in prokaryotes as well as in eukaryotes only a small fraction of all these genes are expressed at a certain state (time step). The proteins encoded in the genes all have to carry out different tasks. As the synthesis of proteins consumes a lot of energy, regulation is indispensable: the cell synthesises only those proteins that are needed for the particular tasks performed at the moment. This ensures that the cell utilises its energy in an optimal way. The frequency of expression called "expression level" of genes is an important unit of measure. Some gene products are permanently needed, e.g. proteins acting as enzymes in the central pathways of the metabolism. The genes encoding those proteins are also called "housekeeping genes". Their expression level is nearly constant. The expression levels of the other genes increases and decreases as a result of reactions on molecular signals and stimuli. An increase of the expression level of a gene is called "induction" of the gene, whereas a decrease is called "repression".

In fact, some genes are regulated as a unit. These units are called "operons".

Genetic Regulatory Networks

In Chapter 2.5, we have described the mechanism of gene expression and protein synthesis. The procedure of gene expression can be regulated at several stages. Apart from the regulation of RNA processing, translation, and the post-translational modification of proteins, the regulation of DNA transcription is the best-studied mechanism [45]. All steps of regulation are carried out by proteins or mRNA produced by other genes. The result is a genetic regulatory network that is structured by complex systems of biological interactions between DNA, RNA, proteins, and other molecules. Figure 5.2 shows an exemplary regulatory network consisting of three genes. The regulatory mechanisms follow the basic principles described below ("Some Basic Regulatory Principles").

The state of genetic regulatory systems such as the system shown in Figure 5.2 depends on the concentrations of its regulatory components (proteins, mRNA). For example, protein B only regulates the expression of gene a when it is present. In practice, it depends on the concentration of protein B whether it regulates the expression of gene a.

Genetic regulatory systems usually provide so-called positive and negative feedback loops. This is important because only feedback loops (especially negative feedback loops) ensure that the systems will not get out of control, i.e. that the expression of specific genes must be cancelled after being accelerated, or the expression level of the "housekeeping genes" must be kept constant. Some systems – especially those modelling the interactive regulation of only few genes – reach a steady state, i.e. a state in which the genes are no longer expressed. Computer-based dynamic modelling of such regulatory pathways aim at simulating the system until it reaches a steady state.

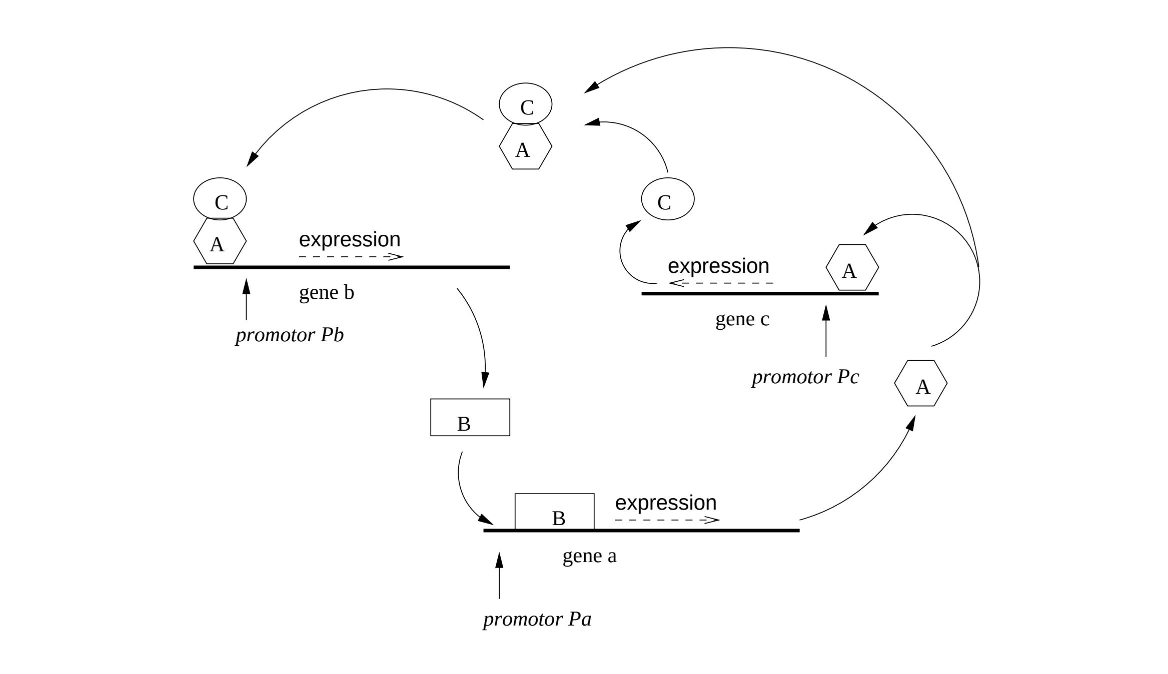


FIGURE 5.2: Example of a genetic regulatory network consisting of three genes. The different steps of the protein synthesis (translation, RNA modification, transcription, protein modification) are not represented. Only the basic interactions between genes and regulatory proteins are shown.

Some Basic Regulation Principles

The regulation of transcription is carried out by protein-DNA interactions. A special enzyme (called "RNA-Polymerase") binds to a specific sequence of the DNA, the so-called *promotor*, and initiates the transcription. All promotors usually have a common sequence, the so-called "consensus sequence", being the optimal sequence for the interaction of RNA-Polymerase and promotor.

A first regulation mechanism mutates one or more base-pairs of the DNA so that the promotor sequence is no longer exactly equal to the consensus sequence. This decreases the functionality of the promotor and thus the value of the expression level. Many housekeeping genes are only regulated by this mechanism.

A second regulation mechanism is based on the influence of regulatory proteins on the interaction between promotor and RNA-Polymerase. Those proteins usually act as repressors (inhibiting the interaction between promotor and RNA-Polymerase), activators (advancing the interaction between promotor and RNA-Polymerase), or other factors (changing the specificity of the RNA-Polymerase for a special promotor). The proteins bind to specific locations close to the promotor on the DNA and provide a negative or positive regulation. The regulation may be mediated through a molecular signal causing the regulatory protein either to dissociate from the DNA or to bind to it.

There are several other regulatory mechanisms but they are too complex and too specific to mention here. Especially eukaryotes provide further mechanisms of gene regulation. See [45] and [53] for further details.

Example: λ Phage Infection of E. coli

The control of gene regulation in the bacterium *E. coli* is one of the best understood systems in molecular biology [33]. We described in Chapter 2.4.2 that cell parasites called viruses can infect a host cell, i.e. that the genetic information of the virus is inserted into the DNA of the host cell because viruses do not have the ability to reproduce their genome on their own.

One of the best studied and thus most frequently modelled genetic regulatory system is the infection of a host *E. coli* cell by a virus called λ phage. We will briefly outline the regulatory pathway of this infection because many simulating approaches take it as a test bed. More details can be found in e.g. [53] or [45].

After infecting an *E. coli* cell, phage λ either propagates as a so-called prophage (lysogenic cycle) or becomes an actively replicating virus (lytic cycle). The decision which cycle to enter results from a complex interaction between the proteins encoded by two genes called *cI* and *cro*. Each gene product acts as inhibitor of the synthesis of the other one.

Lysogenic cycle. The DNA of phage λ is integrated in the genome of the host cell. A high concentration of the product of a gene called *cII* activates the expression of *cI* by stimulating two promotors called p_I and p_{RE}. The product of gene *cII* inhibits the expression of genes that would start the lytic cycle.

Lytic cycle. The lytic cycle is divided into several stages. The first step is called very early stage. Transcription starts at several promotors including p_L, p_R, and p'_R respectively, producing several proteins, among which are also the important N and *Cro* proteins. A high concentration of protein N starts the second stage called early step in which further regulatory proteins are expressed. These further proteins stimulate the synthesis of viral DNA. The *Cro* protein acts as negative feed back loop to regulate the expression of the viral DNA. At the end of the cycle (called the late stage) huge amounts of structural proteins are expressed which are required to build new phages. Beside this, there are also proteins expressed which catalyse the lysis (the damage of the cell wall leading to cell death) of the host cell.

The infection ends in a steady state representing either lysogeny or lysis. Detailed information about the interactions between the according genes, proteins, and promotors can be read e.g. in [45],[53], or [33].

The λ decision circuit controls one phase of a single phage life cycle and operates within a single *E. coli* cell. Many genetic circuits have broader scopes. For example, elements of several cells may be components of a genetic regulatory network that controls the life cycle of different cell types and cell generations. Symbiotic relations between bacteria and higher organisms require the modelling of networks that cross species boundaries [50].

5.1.2 Metabolic Pathways

As already described above, metabolic pathways represent successive biochemical reactions that interconvert molecules within the cell.

Biochemical Reactions

A biochemical reaction is an action where one or more chemical substances are converted into one or more other substances. The chemical components which are interconverted are called metabolites. Figure 5.3 represents a simple chemical reaction. A so-called substrate (S in Figure 5.3) is converted into a so-called product (P in Figure 5.3). "Converted" means that one or more parts of the substrate molecule is rearranged or the substrate is split in two products. The inverse action, where two substrates are merged into one product, is also possible. (See [45] for the five general reaction types in organisms).

$$\begin{array}{lllllll} \text{(a)} & S & \xrightarrow{e} & P & & & \\ \text{(b)} & H_2O & + & O_2 & \longrightarrow & H_2O_2 & \\ \text{(c)} & 2H_2O & + & O_2 & \longrightarrow & 2H_2O_2 & \end{array}$$

FIGURE 5.3: (a) graphical representation of a typical biochemical reaction. (b) reaction of a water and an oxygen molecule into H_2O_2. (c) same reaction representing its stoichiometry

The number of atoms on the left side of a chemical reaction must be equal to the number of atoms on the right side. In fact, there is no chemical reaction that "consumes" atoms. See e.g. the reaction in Figure 5.3 of water and oxygen to H_2O_2. To equalise the number of atoms on both sides of the reaction arrow, one must define that this reaction can only be performed with two molecules of water and one molecule of oxygen. The result will be two molecules of H_2O_2. The so-called stoichiometry of a reaction describes the minimal amount of each atoms that are participating. In general, each chemical reaction in an organism is carried out by an enzyme (e in Figure 5.3).

Biochemical reactions are thus also called enzymatic reaction. In fact, each chemical reaction is reversible. This is true especially for enzymatic reactions. Thus, the reaction formulas in Figure 5.3 should be written with a "$\leftrightarrows$" instead of a "$\longrightarrow$".

Enzymes

Enzymes are proteins that carry out biochemical reactions without being consumed.

Common Properties. The enzymes have several common properties:

- Enzymes can accelerate a reaction. The velocity of a reaction can increase by a factor of 10^{15}.
- Substrate-specificity: Enzymes are very specific to a certain substrate or a certain group of substrates that are similar in three dimensional structure.
- Reaction-specificity: Enzymes carry out only one of the following types of a chemical reaction (see [45] for details):
  - Group transfer: a part of a substrate is transfered to another substrate.
  - Redox reaction: e.g. one H-atom of a water molecule is transfered to a substrate.
  - Isomerisation: the molecules and bonds of one substrate are rearranged.
  - Split: one substrate is split into more products.

Reaction Mechanisms. Enzymatic reactions are based on interactions and usually follow two different types of reaction mechanisms [53]:

- One-substrate-reaction: The enzyme (E) converts one substrate (S). The enzyme interacts with the substrate to build out the (ES)-complex. The enzyme-substrate-complex (ES) is transformed into the enzyme-product-complex (EP). After that, the product (P) dissociates from the enzyme:
$$\mathrm{E+S} \quad \leftrightarrows \quad \mathrm{ES} \quad \leftrightarrows \quad \mathrm{EP} \quad \leftrightarrows \quad \mathrm{E+P}$$
- Two-substrate-reaction: The enzyme (E) converts two substrates (A and B) into two products (P and Q). Generally these reactions follow the mechanism of the one-substrate-reaction by building enzyme-substrate-complexes and enzyme-product-complexes. There are several possibilities of interaction between the enzyme and the substrates, e.g.:
$$\mathrm{E} \quad \stackrel{+\mathrm{A}}{\leftrightarrows} \quad \mathrm{EA} \quad \stackrel{+\mathrm{B}}{\leftrightarrows} \quad \mathrm{EAB} \quad \leftrightarrows \quad \mathrm{EPQ} \quad \stackrel{-\mathrm{P}}{\leftrightarrows} \quad \mathrm{EQ} \quad \stackrel{-\mathrm{Q}}{\leftrightarrows} \quad \mathrm{E}$$

As mentioned already in Chapter 2 the three dimensional structure of the enzyme determines it function. In fact, a special part of the enzyme, the so-called active site, is responsible for the interaction with metabolites. Only a correctly folded protein provides a correctly folded active site which itself interacts only with a substrate whose three dimensional structure fits. Sometimes, additional substances (so-called co-factors) are required to enable the protein acting as desired. In this case the enzyme alone is inactive. Only a complex based on an interaction of enzyme and co-factor is active. In addition, there are so-called multi-enzyme-complexes build up from several distinct enzymes carrying out several specific reactions and providing specificity for diverse substrates.

Regulation

The activity of enzymes must be strictly regulated within the cell to adapt to the cell's requirements, since enzymes carry out nearly all metabolic reactions. There are several ways to regulate the activity of enzymes. We will describe only the very basic principles. An exhaustive discussion will go beyond the aims of this thesis. For detailed information about the regulation mechanisms of enzymes see e.g. [45] and [53].

There are two basic ways of regulation of the enzyme activity:

- The activity of an enzyme depends on the concentration of the enzyme itself. The concentration of an enzyme in the cell is determined by gene regulation (see "Regulatory Pathways"). This is an important intersection of metabolic pathways and genetic regulatory systems.
- In addition, the activity of enzymes depends on the concentration of the substrate(s) and product(s). In pathways that consists of several successive reactions, the enzyme activity often is regulated by the product(s) of a later step (so-called feedback regulation).

The kinetic of enzymatic reaction follows common mathematical laws. We do not describe these laws here because this is beyond the scope of the thesis. [45] and [53] provide detailed information about reaction kinetics.

Reaction Pathways

To build up complex macro molecules out of small substances, the cell has to perform many successive reactions. The sequence of these reactions is called a metabolic pathway. For example the sequence of reactions that synthesize glucose out of pyruvate is called gluconeogenesis-pathway. When we speak of a pathway we mean that a substrate is interconverted via several steps and several so-called intermediates into a product.

Cell Metabolism

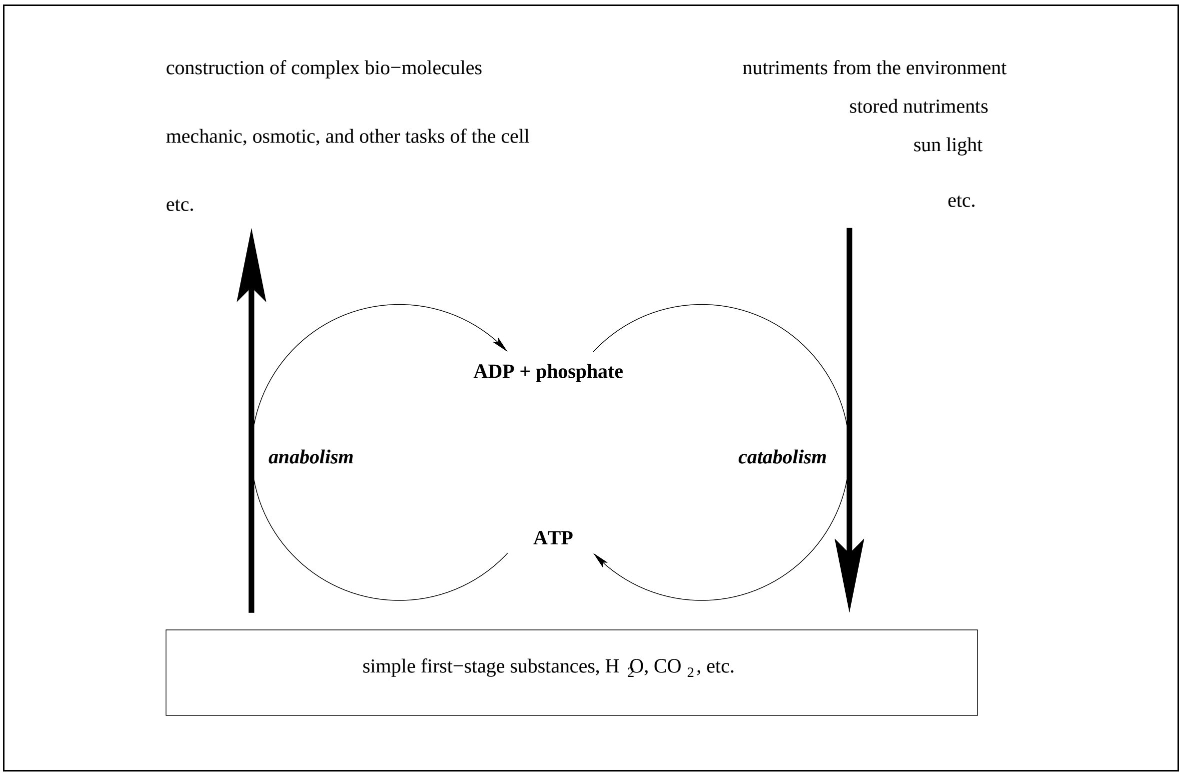


FIGURE 5.4: Overview of the metabolism of a cell.

The metabolism of a cell is the complete amount of all metabolic pathways. We can distinguish between the reactions that build up macro molecules from small chemical substances (these reactions are part of the so called anabolism) and the reactions *vice versa* (these reactions are part of the so called katabolism). The anabolism consumes whereas the katabolism produces biological energy. Within the cell, biological energy is stored in a molecule called ATP. It can be converted into ADP and a phosphate residue.

In Figure 5.5 an overview of the primary metabolism of most organisms is shown. A so-called "ball and stick map" (undirected graph) is used for representation. Balls are intermediates, sticks represent reactions. Enzymes as well as co-factors and small molecules participating (e.g. ATP) are not shown. As one can see, the complete metabolism is a very complex system which is not easy to accommodate.

5.1.3 Biochemical Pathways

As mentioned above, the term “biochemical pathway” is used in many different ways, most frequently as the concept of all kinds of reaction pathways in organisms such as e.g. metabolic pathways, genetic regulatory systems, cell communication, and – very specific – the mitochondrial protein import pathway.

Another frequently used term in this context is *signal transduction*. All chemical pathways in organisms providing the circulation of signals are called signal transduction pathways. In fact, regulatory pathways as well as metabolic pathways are closely related to signal transduction.

In this work we will use the term “biochemical pathway” as the concept that integrates regulatory and metabolic pathways.

Regulatory systems and metabolic pathways usually intersect each other. Consider the following example: A gene a regulates its own expression by encoding an enzyme that catalyses a reaction step in a metabolic pathway. The resulting metabolite M of that pathway induces the regulatory action of a protein P on a.

This simple example demonstrates that regulatory and metabolic pathways are closely related and that models should integrate gene regulation with the metabolism in order to understand the functioning of cells and organisms. (In fact, it also justifies our definition of “biochemical pathways”.)

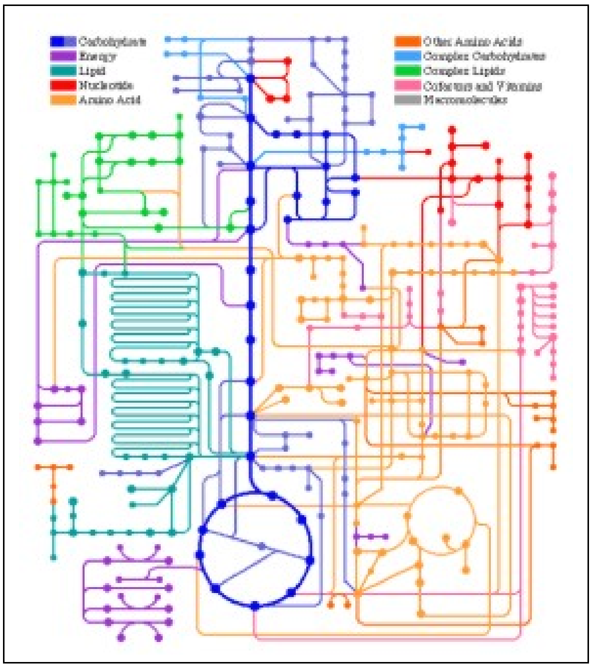


FIGURE 5.5: A ball and stick map (graph) of primary metabolism (taken from [13]).

5.2 Static vs. Dynamic Modelling

Several genome projects are leading to complete characterisations of the genomes of diverse organisms such as yeast, *E. coli*, *C. elegans* (worm), and man (e.g. Human Genome Project). In addition, temporal gene expression patterns are currently being obtained for many cell types allowing to get insight into the complex systems of gene regulation [66].The spatiotemporal expression of genes in organisms is regulated by a system consisting of a large number of genes and their encoded proteins connected through a complex network of interactions [21].

The metabolism interconverts a huge amount of big and small molecules. Its regulation is determined by (among others) interactions between many components and the concentrations of these components.

The behaviours of these systems are hard to understand intuitively. Thus, computer-based modelling and simulation of regulatory and metabolic pathways are most likely indispensable [21]. The computer-based frameworks should be able to use the experimental data on sequencing and gene expression available [66]. They should also be able to deal with the problem that there is only incomplete knowledge about most genetic regulatory systems and also many metabolic pathways available.

Static models of regulatory and metabolic systems yield schematic and pictural representations. Static modelling approaches are not able to deal with time-dependent actions, like concentration changes. They provide qualitative but no quantitative (except stoichiometric) aspects of the model.

Dynamic models of regulatory and metabolic systems try to simulate the quantitative behaviours of the systems.

In the following we will describe some current approaches in modelling biochemical pathways. Each formalism is discussed in the context of regulatory pathways and metabolic pathways.

Modelling and Simulation of Regulatory Pathways. Simulation of genetic regulatory systems is an important, up-to-date problem: To understand the operation of genetic regulatory systems, one need to consider several different issues such as non-linear interactions, positive and negative feedback loops, crosstalk between pathways, and time delays resulting from mRNA or protein transport. Understanding the combined effects of these phenomena is often beyond the intuition [66].

Simulation environments should be able to model several different situations of the cell such as development and growth, differentiation, and generation of appropriated responses to diverse environmental stimuli.

In the following when we are in the context of regulatory pathways, we follow two literature reviews of 2000. These are [21] which is written from the computer science point of view, and [66] which evaluates the usefulness of each approach to model regulatory pathways more from a biologist's point of view.

Modelling and Simulation of Metabolic Pathways. Modelling of metabolic pathways is closely related to the modelling of genetic regulatory networks and also a currently emerging issue. Modelling approaches are used in several different contexts such as among others:

- Static representations are used to accommodate the complexity of the metabolism (see Figure 5.5) and to describe the qualitative aspects, e.g. for teaching and learning.
- Static representations are also used to advance the usefulness of similarity search results. Predicting the unknown function of a sequence S deriving from an organism O, the most common way is to search a sequence database for S. The resulting predictions would be more accurate if they were seen in the context of the metabolism. Let the predicted function of S be the enzyme of a special metabolic pathway P. It would stabilise this prediction if the other enzymes in this pathway would be also present in O. If not so, it indicates that O cannot perform the pathway P and the prediction of the function of S is at least questionable.
- Dynamic simulations are used to calculate the concentration rate chances and predict the behaviour of entire cells.

See [41] for a summary of application areas and challenges.

Knowledge representation formalisms and simulation environments should be able to model biochemical reactions, their components such as enzymes and metabolites, and (in case of dynamic modelling) the according reaction kinetics.

5.2.1 Databases on Genetic Regulatory Systems and Metabolic Pathways

Several molecular biology databases provide data on regulatory pathways such as among others KEGG (cf. Appendix A.52), RegulonDB (cf. Appendix A.93), EcoCyc (cf. Appendix A.25), or ooTFD (cf. Appendix A.73). These databases explicitly model data on regulatory pathways. In general, these models are static. They can be seen as richly annotated graph representations. Tools associated with the databases usually offer facilities to compose networks, visualise networks on different levels of detail, and navigate through the networks.

Many other molecular biology databases provide implicit data, i.e. annotations to the core data (e.g. several entries in GenBank may contain annotations with information about genetic regulation). These data is usually too difficult to obtain to be a basis of genetic regulation modelling approaches.

Metabolic databases usually describe collections of enzymes, reactions and biochemical pathways. Some of them focus on one single organism (e.g. EcoCyc, cf. Appendix A.25), while others provide data that is not specific to any organism (e.g. KEGG, cf. Appendix A.52).

All metabolic pathway databases use the EC number system as primary key or search index. From the biologist's point of view the EC system has several advantages and disadvantages which are described in [40]. Relevant for modelling issues are the facts that the EC system provides unique identifiers for enzyme function but do not cover many types of enzymes in adequate detail.

Another obstacle in the context of naming conventions are the chemical compounds. Multiple names used for chemical compounds cause confusion as biologists often do not use the naming guidelines produced by chemists. As a consequence, extensive synonym lists for enzymes and compounds are indispensable [40].

Databases on metabolic pathways provide static representations of metabolic pathways. Nevertheless, those databases should provide a kind of dynamic representation: to accommodate the complexity of the metabolism, it should be possible to browse through different levels of detail of representation and annotation.

It should be mentioned that similar to databases on regulatory pathways, most other molecular biology databases indirectly and implicitly provide data on metabolic pathways. Those data is often hidden in annotations to the core data.

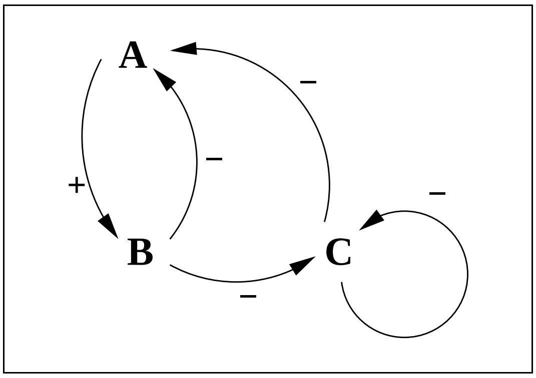
5.2.2 Approaches Based on Graphs

A common way to model biochemical pathways is to use the concepts of directed graphs. A directed graph G is defined as $G = (V, E)$, where V is a set of vertices and $E \subseteq V \times V$ is a set of edges.

Modelling Genetic Regulatory Networks. The vertices of the graph usually represent the genes or other elements of interests in the regulatory system. The edges usually model interactions among the genes.

The graph representation can be specialised or generalised in several ways. Vertices and edges may be labelled to express information about the genes and their interactions, e.g. with either + or - to indicate a positive or negative regulation. The set of edges is defined as a set of tuples $E \subseteq V \times V \times R$, where $R = \{+, -\}$.

Hypergraphs are adequate to model situations in which a cooperation of proteins regulate the expres-



Directed graph $G = (V, E, R)$
$R = \{+, -\}$
$V = \{A, B; C\}$
$E = \{(A, B, +), (B, A, -), (B, C, -), (C, A, -),$
$(C, C, -)\}$

Directed hypergraph $\widehat{G} = (V', E', R')$
$R' = \{+, -\}$
$V' = \{A, B; C\}$
$E' = \{(\{B, C\}, A, \{-, -\}), (\{A\}, B, \{+\}),$
$(\{B, C\}, C, \{-, -\})\}$

FIGURE 5.6: Example: Modelling an exemplary genetic regulatory system with graphs (G) and hypergraphs ($\widehat{G}$)

sion of a gene. A hypergraph $\widehat{G}$ is defined as $\widehat{G} = (V, E)$, where V is again a set of vertices, and $E \subseteq \wp(V) \times V$ a set of multiedges ($\wp(X) = \{N \mid N \subseteq X\}$). Analog to graphs, hypergraphs can be extended by labelling vertices and edges, e.g. by defining $E \subseteq \wp(V) \times V \times \wp(R)$, where $R = \{+, -\}$. This is a common praxis in modelling genetic regulatory systems. The edges are tuples (V_1, v_2, R), were V_1 represents a set of genes regulating the expression of v_2, and R represents a set of signs indicating the regulatory influence (inhibitor or activator) of each element in V_1.

A number of operations on graphs can be computed to retrieve biologically relevant information and to make predictions. Those operations include among others [21]:

- searching for paths between two genes;
- searching for cycles in the network;
- computing global characteristics of the network such as the average and the distribution of the number of regulators per gene;
- searching for loosely connected subgraphs;
- comparing two graphs (e.g. the same networks from different species) for similarities and differences;

Graph models can also be generalised by probabilistic extensions. These approaches are called Bayesian networks (see the following section).

Modelling Metabolic Pathways. The most straightforward formalism to represent metabolic pathways are graphs and hypergraphs. A pathway can be seen as an annotated directed hypergraph. The edges represent biochemical reactions and should be labelled with the enzyme name(s). The vertices are sets of chemical substances, the metabolites. It is also possible that the detailed steps of an enzymatic reaction is modelled using graphs. In that case, metabolites, enzymes, enzyme-metabolite complexes, and other objects of the metabolism are modelled as vertices. the edges model the transitions from one state to the next (e.g. from single-substrate, single-enzyme state to the substrate-enzyme-complex state).

In fact, only few approaches use this formalism to represent metabolic data. There are several formalisms related graph theory that are developed to model metabolic pathways. Usually these models are more efficient to implement and include among others:

- Karp's "predecessor list" [42]
- The KEGG model organising metabolic data in terms of binary relations [30]

5.2.3 Bayesian Networks

Bayesian networks are usually only used to model regulation networks of gene expression. The formalism of Bayesian networks is based on directed acyclic graphs. As mentioned above, it extends graph models by probabilistic aspects.

A Bayesian network is defined as [21]:

- A directed acyclic graph $G = (V, E)$. The vertices $i \in V, 1 \leq i \leq n$ usually represent genes. Each gene corresponds to a random variable X_i describing the expression level of i.
- Conditional distributions $p(X_i|\text{ancestors}(X_i))$ that are defined for each X_i, where $\text{ancestors}(X_i)$ is the list of all random variables corresponding to the direct regulators of i in G (i.e. $\text{ancestors}(X_i) = X_j$, where $(j, i) \in E$).

Both specify a joint probability distribution

$$p(\mathcal{X}) = \prod_{i=1}^{n} p(X_i \mid \text{ancestors}(X_i))$$

A so-called conditional independency $i(X_i; \mathcal{Y} \mid \mathcal{Z})$ expresses the fact that, given $\mathcal{Z}$, X_i is independent of $\mathcal{Y}$, where $\mathcal{Y}$ and $\mathcal{Z}$ are sets of variables. The graph of a Bayesian network usually encodes the so-called Markov assumption $i(X_i; \text{non-descendants}(X_i) \mid \text{ancestors}(X_i))$ for each gene i in G [21].

Two Bayesian networks are defined as equivalent if they imply the same set of independencies. Equivalent networks have the same underlying graphs but may disagree on the direction of some edges [25].

5.2.4 Petri Nets

Another current approach to model biochemical pathways based on graphs uses the formalisms of Petri nets. A Petri net can be defined as $\mathcal{PN} = (P, T, E, W, M_0)$, where

- P is a finite set of places $p_i \quad 1 \leq i \leq n$
- T is a finite set of transitions $t_i \quad 1 \leq i \leq n$
- E is a set of arcs $\quad E \subseteq (P \times T) \cup (T \times P)$
- $W : E \rightarrow \mathbb{N}$ is a set of arc weights
- $M_0 : P \rightarrow \mathbb{N}$ is the initial marking

Note that P and T are disjoint sets.

$\mathcal{PN}$ is a graph formed by two kinds of nodes (places and transitions) and directed edges (arcs). A non-negative integer number ("token") is assigned to each place and can vary depending on the state of $\mathcal{PN}$. A positive non-zero integer is assigned to each arc specifying its "weight".

A transition ("event") is enabled when the number of tokens in its input places is greater than or equal to the weights on the arcs connecting the input places to the transition. A transition with no input places is always enabled. An enabled transition can fire: the number of tokens determined by the arc weights is deposited in the output places.

The state of $\mathcal{PN}$ representing the number of tokens present in each place is called "marking". The firing of a sequence of transitions may change the marking of $\mathcal{PN}$. M_o is the initial marking of the petri net.

Modelling Genetic Regulatory Networks. Ordinary Petri nets (as defined here) allow discrete modelling of genetic regulatory systems. Places may represent genes and other components of the system, whereas transitions model the interactions between the genes.

Modelling Metabolic Pathways. The formalism of petri nets is commonly used to model a metabolic network as a discrete-event system [59]. Places of a petri net would represent compounds participating in a bio-reaction, such as metabolites, enzymes, co-factors, etc. One place represents one compound but it is also possible that two or more places represent the same compound according to the physical attributes or distinct functions (e.g. one place for the in-active form of an enzyme, and one for the active form). Transitions represent individual reactions or a series of forward reactions (in case that the intermediary compounds are not of interest). Tokens indicate whether a compound is present or not and arc-weights represent the stoichiometry of reactions.

5.2.5 Boolean Networks

Another approach which is usually only used to model genetic regulatory networks is the approach of Boolean networks. The state of each gene is represented by a Boolean variable. This variable tells whether the gene is active (on, value of the variable is 1) or inactive (off, value is 0) and hence whether the products of the genes are present or not. Interactions between genes can be modelled using Boolean functions calculating the state of a gene from the activation of other genes. The result is called Boolean network.

All variables of the Boolean network are represented using a vector $\vec{x}$ (in case of n elements in the regulatory system, $\vec{x}$ is a n-vector). Each x_i has the value 1 or 0 representing whether gene i is active or inactive.

The value of the Boolean variable x_i (called the state of x_i) at time-point $t+1$ (written $x_i[t+1]$) is computed by a Boolean function f_i (also called rule). The rule f_i computes its output (state x_i at time-point $t+1$) from the state of $k_i \leq n$ elements at the previous time-point t (input). Note that k_i may be different for each x_i.

The dynamics of a Boolean network modelling a regulatory system can be computed as follows [21]:

$$x_i[t+1] = f_i(\vec{x}[t]),\ 1 \leq i \leq n,$$

where f_i computes an output value from k_i inputs.

Usually, the transition from one state to the next is determined in a parallel fashion. Translations between states in a network are deterministic (each Boolean function generates a single output from a given input) and synchronous in the sense that the outputs of the elements are updated simultaneously.

Several computations that can be performed on a Boolean network, e.g. on its state space, can yield to predictions and interpretations of local properties for the global dynamics of the network.

5.2.6 Rule-based Approaches

Another approach that is suitable to model biochemical pathways, especially if there is not much quantitative knowledge about the system to be modelled, are knowledge-based or rule-based formalisms.

Modelling Genetic Regulatory Networks. In general, rule-based formalisms to model regulatory systems consist of two components [21]:

1. A set of facts that express knowledge about the objects of the system. Each object is described by properties (attributes). The objects are usually structured in a class-hierarchy.
2. A set of rules that are usually stored in a database. Rules consist of two components: First a condition part expresses conditions in terms of properties of objects. Second an action part operates on the objects (e.g. changing a property of an existing object).

Modelling Metabolic Pathways. Rule-based formalisms usually model biochemical pathways in terms of reaction rules. Such a metabolic rule r could be defined as follows [34], [35]:

$r = (S, P, E, I)$ where

- S is a multi-set of preconditions representing the substrates of a biochemical reaction,
- P is a multi-set of preconditions representing the products of a biochemical reaction,
- E is a multi-set of catalyzed conditions representing enhancers of the reaction, and
- I is a multi-set of catalyzed conditions representing inhibitors of the reaction,

A metabolic pathway P can then be modelled as $P = (C, R, c_0)$ where C is a set of configurations, c_0 is the start configuration, and R is a set of metabolic rules.

A rule will be activated if the elements of its component S are elements of the actual configuration. The action of a rule r will modify the configuration e.g. in the following way [34], [35]: All elements of the actual configuration that are elements of S in r will be removed. All elements of P of r will be added.

Rule-based simulation is based on the repeated process of matching the facts or configurations against the condition parts of the rules. If the conditions of a rule are satisfied the action part of the rule is carried out. Some conflicts that arise when several rules match at the same time have to be resolved by a dedicated control strategy determining the order in which the rules are evaluated. Those control strategies vary from simple random selection to complex priority schemes [21].

5.2.7 Differential Equations

The most widespread formalism to model dynamical systems such as biochemical pathways are differential equations. In this approach concentrations of RNAs, proteins, metabolites, and other elements of the system are modelled by time-dependent variables which values are non-negative real numbers. Interactions between the components of the system are represented by functional and differential relations between the according (concentration) variables. So-called "reaction-rate equations" express the rate of production of a component within the pathway.

Usually the reaction-rate equations are non-linear, thus an analytical solution of the reaction-rate equations is not possible. In special cases only, qualitative properties of the solutions can be obtained [21].

There are currently two strategies to overcome this obstacle: First, the models can be simplified so that interactions are modelled using a set of piecewise-linear differential equations instead of ordinary non-linear differential equations. Piecewise-linear differential equations have mathematical properties that facilitate their analysis.

Second, one can approximate the solutions of non-linear differential equations using numerical techniques. This approach is called numerical simulation. There are several tools that specifically carry out simulation and analysis of biochemical reaction systems such as GEPASI [52]. Numerical simulation is often complemented by techniques from system dynamics, e.g. so-called "bifurcation analysis" tools (coming from non-linear dynamics and chaos research) to investigate the sensitivity of steady states [21].

Modelling Genetic Regulatory Networks. Usually, reaction-rate equations have the form [21]:

$$\frac{dx_i}{dt} = f_i(\vec{x}) \qquad i = 1, \ldots, n$$

where $\vec{x}$ is a vector of concentrations of all elements that have a regulatory influence on gene i, and f_i usually is a non-linear function.

Note: Approaches based on differential equations as described here implicitly assume that the regulatory systems of interest are spatially homogeneous (i.e. there is no input from the environment of the system and also no output). But these assumptions are not appropriate in some systems such as systems that model inter-cell communication. In cases where multiple compartments of cells or multiple cells are needed to be modelled explicitly, more complex differential equations have to be constructed.

Modelling Metabolic Pathways. Quantitative, dynamic simulation of the metabolism is in general based on differential equations. We will show how an exemplary metabolic pathway can be modelled using differential equations:

Consider a simple linear pathway $\mathcal{P}$:

$$S \quad \underset{e_1}{\overset{v_1}{\leftrightarrows}} \quad X_1 \quad \underset{e_2}{\overset{v_2}{\leftrightarrows}} \quad X_2 \quad \underset{e_3}{\overset{v_3}{\leftrightarrows}} \quad P$$

where S is the pathway substrate, P is the pathway product, X_{1-2} are the pathway intermediates, v_{1-3} are the reaction rates for the individual steps, and e_{1-3} are the enzymes of the according reaction.

$\mathcal{P}$ can be modelled by the following equations:

$$\begin{aligned}\frac{\delta S}{\delta t} &= -v_1 \\ \frac{\delta X_1}{\delta t} &= v_1 - v_2 \\ \frac{\delta X_2}{\delta t} &= v_2 - v_3 \\ \frac{\delta P}{\delta t} &= v_3\end{aligned}$$

In addition, the kinetic relationships can be specified, e.g. using Mass action kinetics [13]:

$$\begin{aligned}v_1 &= k_1 S - k_{-1} X_1 \qquad (= v_{\text{forward}} - v_{\text{reverse}}) \\ v_2 &= k_2 X_1 - k_{-2} X_2 \\ v_3 &= k_3 X_2 - k_{-3} P\end{aligned}$$

where k_x are the rate constants as determined by the use of Mass action kinetics.

Dedicated tools such as GEPASI [52] can simulate the concentration rates of X_{1-2}, S, and P.

5.2.8 Simulation Environments for Biochemical Pathways

Several of the current molecular biology databases integrate data on regulation of gene expression and metabolic pathways, e.g. KEGG (cf. Appendix A.52) or EcoCyc (cf. Appendix A.25). These databases are based on static modelling and representation.

Few approaches aim at dynamic modelling and simulating the behaviour of an entire organism. Current approaches yield simulation environments that dynamically model the behaviour of simple cells and viruses. In general, these approaches are based on differential equations. In the following we will describe three exemplary simulating tools that aim at integrating metabolic simulation and gene expression simulation. A short discussion compares these tools according to their modelling power and try to suggest some general thoughts.

Note: In the following, we use the term “target system” for the biological system to be modelled using the simulation tool.

The E-Cell Project

E-CELL [68] is a generic environment for simulating molecular processes in user-definable models. It integrates modelling and simulation of diverse cellular processes such as:

- gene expression, and
- metabolic pathways and signal transduction (biochemical kinetics).

E-CELL is a rule-based simulation tool written in C++, that models all aspects of the target system with differential equations. The model of the target system, which can be defined by the user, consists of three lists (input files):

- The “substance list” defines all objects within the target system.
- The “rule list” defines all reactions and their kinetics.
- The “system list” defines the spatial and/or functional structure of the target system.

The state of the simulation system is represented by a list of concentration values of all substances defined (“substance list”), together with global biochemical values (such as cell volume, temperature, etc.). The next state in time is generated by computing all functions defined in the “rule list” as differential equations. For simulation of the model, numerical integration methods are used. These numerical integration methods are implemented in object classes. As a consequence, the user can decide which method to use for which equation groups (e.g. all gene expression equations) and also add a new method by adding an according object class.

The graphical user interface of E-CELL provides functionalities for defining user-specified models, for interacting with the simulation (e.g. change environmental conditions), and for visualising the results of the simulation. The latter one is done by displaying the concentration values of each substance along a time axis.

Testing of the E-CELL simulator was carried out modelling a hypothetical, minimal cell based on the gene set of *M. genitalium*, the organism with the smallest genome (580kb) known [68]. The gene set of *M. genitalium* was reduced to only 127. Only a partial set of metabolic pathways of *M. genitalium*, is modelled. The simulating tool produced satisfying results, such as intuitively predicted reactions on environmental changes, indeed, “under unrealistically favourable environmental conditions” [68].

The tool was not yet tested for modelling regulatory pathways although [68] state that “the software itself allows the user to write rules for sophisticated gene regulatory reactions”. Complex signalling pathways are also not yet implemented and thus not yet tested. The authors [68] suggest that the equations used in E-CELL model signalling pathways accurately and satisfyingly.

The software E-CELL can be obtained at the project’s homepage at: `http://www.e-cell.org`

The Virtual Cell

The Virtual Cell of the National Center for Research Resources (NCRR) [61] is a tool for cell biologists to create models and to produce predictions from the models via simulation. The user can specify models using biological abstractions. Each model consists of a collection of so-called “species” (chemical substances), “reactions” (biochemical kinetics), and “features” (geometric information of the simulation environment). In addition, spatial and environmental information is taken into account for simulation.

The tool automatically generates code to implement the specified model. All reactions are represented through differential equations. Numerical simulation is used to solve these equations.

The Virtual Cell can simulate single cell compartments as well as entire cells consisting of several compartments. The simulator engine consists of two components. The simulation controller is running together with a database server for storing the user specified models on a UNIX server. The graphical user interface is implemented as a Java-Applet.

Details of the mathematics and numerical methods The Virtual Cell is using for simulation can be found on the project’s web site at: `http://www.nrcam.uchc.edu/`

BIODRIVE

BIODRIVE [44] is a tool for simulating a user defined biochemical model. Users are able to model and study the following processes within a single cell:

- gene expression,
- metabolic pathways and signal transduction (biochemical kinetics), and
- cell-cell interactions (multi-cellular functions).

BIODRIVE requires two input files. First, the so-called "rule file" defines the model of the target system to be simulated, in particular the substances, reactions, and environment of the target system. Second, the so called "initialise file" defines the initial conditions of the substances and thus the state of the system at start of simulation.

The simulating tool consists of a simulator kernel and a graphical user interface (GUI). The kernel translates the "rule file" into the mathematical model. All models – including models for biochemical reactions, gene expression, diffusion, and ligand-receptor interaction – are based on differential equations. After initialising the model according to the "initialise file", the kernel computes the equations and uses numerical integration methods to solve them. The GUI displays the concentration of substances along a time axis or along a spatial axis.

BIODRIVE was tested successfully modelling the "smad signal transduction" pathway of *Drosophila* (fruit fly) yielding satisfying results [44].

To address the stochastic aspects of gene expression, the authors [44] plan to implement stochastic kinetic models and to integrate an optimisation algorithm to determine ideal parameter sets for the differential equations.

The project's web site can be found at: `http://www.symbio.jst.go.jp`

5.3 Discussion and Perspectives

In the following, we will discuss the advantages and disadvantages of the approaches for modelling biochemical pathways described above. We also try to give some research perspectives.

5.3.1 Discussion

The discussion follows [21] and [66].

Databases on Biochemical Pathways

The number of databases providing knowledge on genetic regulatory systems are slowly growing. This is necessary to overcome the data bottleneck that is currently one of the highest hurdles in modelling and simulating genetic regulatory networks. At least, these databases explicitly model regulatory interactions.

Current databases providing metabolic knowledge still have weakness in representing pathway networks. The accurate and clear representation of the complexity of the metabolism is the main challenge in this context. Browsing facilities are (if not missing at all) not yet mature. Modelling approaches based on XML might improve these facilities (see below). In addition, metabolic databases model the metabolism on a rather high level. Data on interactions for between enzymes and intermediates or on reaction regulation and kinetics are usually missing. In the future, this kind of data should be integrated. Simulation environments are not yet associated with the databases.

Approaches Based on Graphs

The approaches of using graphs to model biochemical pathways are probably the most intuitive and straightforward ones. Drawing and other algorithms can be adapted from graph theory to investigate the qualitative character of a biochemical pathway. Although graph models are not able to simulate the quantitative and dynamic aspects of a pathway, they are important tools for teaching and learning because they explicitly model the static aspect of pathways.

Bayesian Networks

The solid basis of Bayesian networks in statistics is profitable to deal with the stochastic aspects of gene regulation. Another advantage of this approach is, that it can be used even if only incomplete knowledge about the system is available. The great disadvantage of the usage of Bayesian networks for modelling regulatory pathways is that they leave dynamic aspects implicit. Like graph approaches, Bayesian networks explicitly model only static aspects. A possible generalisation of Bayesian networks to model also dynamical aspects are so-called dynamical Bayesian networks (see [21] for references).

Petri Nets

The formalism of ordinary Petri nets can be used to develop discrete models of biochemical pathways. A specialisation for the modelling of regulatory networks, where the number of tokens deposited at each place is either 1 or 0 and the weight of each arc is 1, is closely related to boolean networks (genes are modelled to be fighter on or off). But Petri nets cannot represent continuous values such as concentrations of mRNA, proteins, enzymes, or metabolites. In this context, [49] proposes a hybrid Petri net approach, extending ordinary Petri nets to handle continuous values. Places and transitions are defined to be either discrete or continuous. Discrete places have a non-negative integer number of token associated with it. Continuous places are associated with a non-negative real number of token. Continuous places may now model concentrations of the system components. A variable is assigned to each transition. In case of a discrete transition this variable determines its delay time, where as in case of a continuous transition the variable represents its speed. [49] uses this extension to model genetic regulatory networks and states that this extensions allows one to “express explicitly the relationship between continuous values and discrete values while keeping the characteristics of ordinary Petri nets soundly” and that “stochastic factors can be included properly”.

Boolean Networks

Because of their strong simplifying assumptions Boolean networks enable efficient analysis also of large networks. But the simplifying assumptions cause several problems, two of which are mentioned here: First, a gene is modelled to be either on or off. But in reality, genes have also intermediate expression levels that are neglected by the model. Second, transitions between activation states of the genes are modelled synchronously. In reality, this is not the case. Both problems may cause that certain behaviours of the system are not predicted. To overcome these problems [21] mention more generalised logical formalisms, e.g. approaches allowing the variables in the Boolean network to have more than two values and transitions between states to occur asynchronously (see [21] for references). But [66] state that at least asynchronous transition systems solve these problems only partly. In addition, time delays resulting from macro molecular transport often cannot be modelled accurately using a Boolean approach [66].

Rule-Based Approaches

The major advantage of rule-based formalisms is their capability to deal with a richer variety of biological data [21]. Rule-based approaches provide an intuitive formalism to model biochemical pathways. Dynamic behaviour of pathways can be studied, but usually only in a discrete way. Analysis of the dynamic behaviour in terms of continuous concentration rate changes is not possible. Nevertheless, the rule-based formalism has a major advantage. The metabolic rules can easily be adapted from existing knowledge bases. Translating a given metabolic reaction into the rule-based formalism is carried out as follows: All substrates of the reaction are added to S, all products are added to P, and so on. Pathway data in molecular biology databases is often represented in such a way that this translation can be done automatically.

On the other hand, it is difficult to maintain a consistent knowledge base of biological data. Moreover, the problem of integrating symbolic and numerical knowledge in a single formalism is not yet solved satisfyingly.

Differential Equations

Differential equations are the most widespread formalisms to model biochemical pathways. This approach allows modelling of gene regulation and metabolism in great detail. But one serious handicap is that the information required to build such models is often lacking (especially in case of genetic regulatory systems). For numerical simulation techniques to approximate solutions for the equations, values of numerical parameters are available for only a handful genetic regulatory systems or only for parts of systems. Several researchers such as [66] and [21] hope or suggest that these problems could be solved by the growing availability of gene expression data together with new numerical techniques. In case of metabolic pathways, the theory of reaction kinetics based on several different concepts (such as among others Mass action kinetics, Hill kinetics, or Michaelis-Menten kinetics [13]) is well investigated but in practice experimental data required to calculate numerical simulation are also often lacking.

A further problem of the approach using differential equations for modelling genetic regulatory systems is that it makes several implicit assumptions. The modelling approach presupposes that concentrations of substances vary continuously and deterministic. From the biologist's point of view this

TABLE 5.1: Comparison of the modelling capabilities of the three simulating tools described in this section.

Simulator	metabolic pathways	gene expression	signal transduction	others
E-CELL	yes	yes	yes (not yet tested)	
Virtual Cell	yes	yes		
BIODRIVE	yes	yes	yes	cell-cell interactions

assumptions are at least questionable. [21] proposes more discrete and stochastic models based on differential equations to deal with this problem.

Simulation Environments

Most approaches in modelling biochemical pathways focus on the simulation of the phenomenon under study or characterise single physiological mechanisms. The modelling of a new phenomenon requires a new simulation program to be written. This is a time consuming and error prone exercise [61].

In fact, this suggests that a general framework for modelling and simulation of user defined systems is required. The three tools described in this section are a small selection of attempts to fulfill this requirement.

Thus, an important criterium for discussion is the modelling power of each tool. This determines usefulness of the framework from the users' point of view. The modelling power of the three tools described above is summerised in Table 5.1.

Of course, model management is much more useful than implementing one fixed model. But a huge hurdle for the easy usage model managers and simulators such as the approaches described here is that there is not yet any standard format to build and describe a model of a target system. Another obstacle which should not be neglected is that biologists are usually not equipped with sufficient mathematical, physical, or computational expertise. Therefore, it might be desirable to hide the mathematical complexity of the model from the user.

Beside all those problematic thoughts, it should be mentioned that such simulation frameworks greatly benefit the daily work of biologists. A prediction based on the simulation of a model is often most useful if it doesn't match the experimental details because this indicates that the model is not complete or contains errors. Such results often support biologists in searching for new interactions by pointing to the right direction.

Summary

Modelling Regulatory Pathways. Modelling regulatory pathways aims at static representation and/or dynamic simulation. Recent attempts to simulate regulatory pathways can be divided into boolean and continuous approaches.

Boolean approaches assume [66]:

1. the state of each gene can be characterised as either on or off,
2. the regulatory control of gene expression can be approximated by boolean logical rules,
3. all elements are most commonly assumed to update their states synchronously.

The continuous approach usually uses differential equations and requires the assumption of specific kinetic schemes [66].

Hybrid approaches try to integrate continuous modelling based on differential equations within the framework of discrete qualitative modelling. An interesting hybrid approach is described in [49]: The formalism of petri nets is extended to handle the continuous values of genetic regulatory systems such as concentrations of gene products. This approach is promising because it integrates continuous and discrete aspects within on single formalism.

Modelling Metabolic Pathways. The most commonly and frequently used way to represent metabolic pathways in static pictures are annotated graphs. To accommodate the complexity of the metabolism these representations should be able to view single pathways as well as the entire metabolism on different levels of detail. A promising approach is the www interface of the KEGG database providing the capability to browse through pathways of different detail and navigate to annotations. Dynamic modelling approaches can be divided into two categories (see also [29]):

- Stoichiometric models: these approaches allow the study of the dynamic behaviour of metabolic pathways but are limited to discrete simulation. The simulation results of stoichiometric models are predictions such as “how many molecules of substance X are produced under a given starting configuration” rather than changes of concentrations of the enzymes and metabolites.
- Kinetic models: these approaches are in general based on differential equations. The dynamics of metabolic reactions are described by these equations in combination with the known stoichiometry of the reactions. Note that this modelling approach is only appropriate when detailed information about the kinetics of the specific metabolic processes are available.

5.3.2 Perspectives

Static Modelling. Biochemical pathways have to be modelled for static representation. Clearly arranged graphs are e.g. suitable knowledge representations of existing pathway data. These static representations are reasonable tools for education, teaching, and reviewing.

Molecular biology databases provide a huge amount of pathway data but only few of them model and provide these data explicitly. This is an unfortune situation: Dynamic modelling of regulatory systems is currently hampered by a data bottleneck. Quantitative and qualitative data is lacking to develop mature models. Molecular biology databases focusing on metabolic and regulatory pathway data could play an important role as information infrastructures to overcome this bottleneck (see also below).

Note, that static representations of pathway data such as pathway graph diagrams should provide a kind of dynamic behaviour which is different to the dynamics of simulating systems. Especially molecular biology databases focusing on pathway data should provide capabilities to navigate through the data representations in diverse ways. The KEGG (cf. Appendix A.52) database on metabolic and regulatory pathways is a prominent example. The user interface of KEGG provides capabilities to view pathways on different levels of detail and browse through these different views by hypertext link navigation. In addition, the KEGG interface enables the user to browse through annotated information about single components of each pathway. This includes among others e.g. specific information about an enzyme in a metabolic pathway.

The possibilities of dynamic navigation through pathway data are not limited to those mentioned here. Moreover, they depend on user requirements. But also the existing navigation capabilities do not pose optimal solutions. These thoughts yield to the suggestion that XML could be once again a good candidate for modelling biological data – here pathway data – because of its powerful linkage concept (XLink [73]).

Dynamic Modelling. "Modelling of stochasticity and of alternative biochemical architectures is far from complete" [66].

Current modelling and simulation studies are mostly restricted to regulatory systems of modest size and complexity that have been well-characterised already by experimental results [21]. The analysis of larger systems integrating diverse biochemical pathways such as regulatory and metabolic pathways is of great theoretical and practical relevance but is currently hampered by several obstacles, including among others [21], [66]:

- Necessary information for building models, estimate parameters, and testing predictions is not yet available for many regulatory systems.
- The inherent complexity of optimisation problems like estimation of model parameters and computing numerical solutions for differential equations drain and overextend computational resources rapidly even for relatively small systems.
- Usually several simplifying assumptions have to be made to develop an efficient model and system. The price being paid for these assumptions is a potential loss of biological relevance of the simulation results.

The emergence of new experimental techniques in molecular biology will produce more data on regulatory systems. Together with the development of databases and other information infrastructures making experimental data available this will probably overcome the lack of data.

In addition, constantly increasing computer power will allow new approaches in modelling and simulation to be tried.

In general, computer-based simulation and laboratory experimentation have to work hand-in-hand to generate more predictions on responses of cells and organisms to physiological stimuli as well as new data about pharmaceutical agents used in drug discovery and gene therapy. A tight cooperation of both computer tools and experimental work to support the construction of dynamic models for biochemical pathways is currently gaining increasing attention [21].

Last but not least, we want to deal with the question posed in Chapter 3 whether the Object-Protocol Model (OPM) could be adequate for modelling metabolic and/or regulatory pathways.

OPM seems to be adequate for static modelling. Reaction steps could be modelled as sequences of subprotocols, whereas the pathway would be the superprotocol. This could be an interesting approach since OPM provides object-oriented constructs to model further details such as information on chemical components, or information on interactions between the pathway components.

On the other side, OPM is most likely not suitable to model metabolic pathways for dynamic simulation. As far as we know, OPM provides constructs to model time-dependent aspects but no constructs to model quantitative aspects of actions nor constructs to simulate the changing of continuous values along a time axis.

Chapter 6

Conclusions

In this Chapter we try to summerize and discuss the primary results of the thesis.

Molecular biology is a rapidly evolving, highly data-intensive discipline. The usage of computer science techniques for storage, retrieval, and analysis of biological data is indispensable.

Our aim was to introduce the computer science generalist to modelling of biological data. Since most computer scientists do not have a profound expertise in molecular biology we introduced the basic concepts of this domain. Our emphasis was on molecular biology databases and their applications. We tried to provide an overview of the state of art in molecular biology databases and molecular biology database integration. A list of 120 databases in Appendix A provides with a snapshot of currently available data sources in biological science.

Furthermore, we introduced into modelling of biological relationships and processes. We overview current approaches in modelling genetic regulatory systems and metabolic pathways.

6.1 Summary

6.1.1 Molecular Biology Databases

The number of molecular biology databases is steadily growing. The primary reason for this is the heterogeneity of the data and aims of the different databases. While some databases focus on data on a single organism, others try to provide data on specific biological entities (such as nucleic acids or genes). We classified the state of art of molecular biology databases according to the underlying data model and data management system, query interfaces, data acquisition, and contents.

Data model and data management. Molecular biology databases use the different approaches for data representation and data management. The spectrum ranges from

- databases using standard DBMS (relational/object models),
- databases using OPM on top of a relational DBMS,
- databases using ACEDB as DBMS, and

- databases implemented as flat files (explicitly structured through indexes called “line types”, or implicitly structured text files).

Flat files have emerged to be the *de facto* exchange standard in molecular biology data flow. As a consequence many commonly used analysis tools in molecular biology work on flat files.

Data retrieval. Molecular biology databases offer the following forms of data retrieval:

- query interfaces which can be classified either as
  - *ad hoc* query interfaces (supporting free-form queries in a query language) or
  - free-form query interfaces (usually providing with a view on the database),
- facilities to download the entire database or parts in flat text files, and
- facilities to browse through hierarchies of the data which are built by the database providers.

The most common way to retrieve data from molecular biology databases is to download flat text files, because flat files are the standard exchange format and analysis tools require input in flat file format (as mentioned above).

Data acquisition. Usually molecular biology databases aim at being as comprehensive and up-to-date as possible. To achieve this goal, data acquisition plays a central role.

In general, molecular biology databases acquire their data from

- the research community (research teams and institutes that submit their data to the databases),
- the scientific literature (publications in journals, books, etc.), and/or
- other databases.

Several problems specific to the acquisition methods have been mentioned and discussed, including among others: overhead costs, waste of time and human capacity, data format conversions, etc.

6.1.2 Molecular Biology Database Integration

Biological data are scattered over many different stand-alone sources. The incompatibility and heterogeneity of these sources are strong limitations to the use that can be made of the available data. An integrated information infrastructure for molecular biology is needed to gain full profit from the available data.

Several hurdles must be overcome when creating an integration environment for molecular biology databases. In this thesis, we described requirements and key problems in that context such as heterogeneity on the semantic, query language, and data modelling level and the difficulties of updates.

We further overviewed and classified a selection of current approaches in molecular biology database integration and identified the following categories:

- theoretical approaches (ontologies and standardised exchange formats),
- navigation through hypertext links,
- data warehouses,
- multi-database querying systems, and
- federated database systems.

In addition, the integration systems are classified either as warehousing or federated approaches.

6.1.3 Modelling Biochemical Pathways

Beside the summary of data models that represent static biology data in molecular biology databases we described current approaches for modelling a special form of biological interactions, genetic regulatory systems and metabolic pathways, in this thesis referred to as biochemical pathways. We gave an overview and discussed formalisms currently used to model biochemical pathways, including:

- approaches based on graphs,
- Bayesian networks,
- Petri nets,
- boolean networks,
- rule-based formalisms, and
- differential equations.

We classified these approaches in static and dynamic modelling.

6.2 Discussion

Molecular biology distinguishes between the *genotype* and the *phenotype* of an organism. The genotype is determined by the genomic particularity of the organism, its genome. The phenotype is characterised by the phenomenon and actions of the organisms determined by the genotype and the environment. A demonstrative analogy is the following, given by [69]: Consider a complex computer tool. The code of the software (assuming it will never be changed) would be the genotype whereas the dynamic execution behaviour of the system, depending on the code, interaction with users, the operating system, etc. would be its phenotype.

Biologists investigate both the genotype and the phenotype of organisms. They thus produce genomic data as well as data on phenotypes ranging from gene products, complex interactions between these gene products to the behaviour of entire organisms.

In deed, the amount of genomic data which are currently existing and are being produced in the future is just overwhelming. But one major challange that molecular biology and bioinformatics face is the interpretation of this information in terms of the biological function of the genes and proteins encoded in the genome – with other words the generation of phenotype data. Most likely, this interpretation will lead to an amount of data that is a good deal more than the amount of genomic data. It is clear that all these data can only be conquered with the support of computers.

Biological data have a characteristic structure that is different to other domains. While genomic data is represented in a well-known way through sequences over the four-letter alphabet $\{A, C, G, T\}$, there is no clear model for data on the phenotype of organisms. For example, the expression products of genes, the proteins, can be represented as sequences over a 20 letter alphabet, but this representation is so simple that it allows only very few insights into more important properties such as three dimensional structure of proteins.

Reasons for the fact that there is no concrete model for phenotype data may be:

- Different analysis and experimental methods generate differently structured data sets.
- The situation where all biological entities and their relationships and interactions are determined in full detail is far out of reach.
- The notion of biological function differs according to the aims of research. Different researchers having different notions focus on differnt aspects of biological entities and thus produce semantically and structurally different data sets.

There are several consequences: First, the structures of single data sets and database models are rapidly evolving. Earlier attempts to standardize data formats have failed and future attempts will also shipwreck. Standard DBMSs usually cannot adequately deal with this fact. Second, data are highly heterogeneous not only in structure (syntax) but also in their meaning (semantic). Generic ontologies are good candidates to address this problem. Third, biological data are a prototype for what can be referred to as *semistructured data* [2]. Biological data often is incomplete and different data sets on the same biological entity often do not fit into a common model. Standard DBMSs are again not adequate to deal with this fact. For example relational DBMSs are inefficient since one has to maintain and explore many quasi empty tables. Frequently occurring schema changes quickly lead to complex and unintuitive database schemas.

Therefore, it is perhaps no wonder that the exchange standard in molecular biology are flat files. These flat files consits of data items called entries. Each entry usually consists of all information available on a semantic item. This is the result of attempting to achieve human readability: all information associated with one entry are kept together. As a consequence, each data repository has to provide a flat file version of its contents. Analysis tools require input as flat files (often different tools require differently structured files).

Flat files as standard data format have several disadvantages. First, they are difficult to use because they require parsing tools which are complex to write and must be adapted to every single schema change. Second, they waste computer resources because databases have to store the entire data in several different formats or have to produce several different formats by converting the original data set (to address the fact that most analysis tools require input in different formats). Last, the goal of normalisation and no redundancy is disregarded (e.g. the classification of an organism might be repeated in all entries of that organism).

The development of flexible formalisms for knowledge representation that can deal with the domain-specific challenges and are capable to model different notions of biological function is needed. We propose the use of XML because of its close relationships to documents (especially web documents), its ability to address the semistructured flavour of biological data, and its power to cross-link related information. XML can also play a central role in molecular biology database integration.

Appendix A

Directory of 120 Selected Databases

We selected 120 databases to be described in this section in some detail. This selection is close to that of the "Database Issue 2000" of the journal Nucleic Acids Research (NAR, 2000, Vol. 28, No. 1). Some randomly selected databases have been added. Others, mentioned in the "Database Issue 2000" were left out. The order of appearance is alphabetical according to the abbreviations (if one is provided) of the database names. With our selection we try to cover the whole spectrum of computer science techenologies used for developing molecular biology databases.

It should be pointed out, that this is just a "snapshot" as of Spring 2001. We focus our description on the classification criteria mentioned in Chapter 3.2. A grand table, summerising this classification is presented in Chapter 3.5.

A.1 3DBase

Provider and access. 3DBase was developed in 1996 at the Brookhaven National Laboratories but as the PDB migrated to another location, links to 3DBase are no longer provided. Nowadays the 3DBase seems to be integrated in the PDB (cf. A.75) and is accessible through the so called 3DBrowser. The 3DBrowser is e.g. provided by the Weizmann Institute of Science, Israel at: `http://pdb.weizmann.ac.il/pdb-bin/pdbmain`

All information provided here refer to the original development of 3DBase at Brookhaven.

Contents. 3DBase provides the structural information stored in the PDB (cf. A.75). The complexity of PDB entries and their use by a multi-disciplinary community required the construction of a database that representsstructural, biological, chemical, and bibliographic information.In addition to all coordinate entries found in PDB, the database contains semantic links to entries found in other databases. For example, 3DBase represents the relationships between sequences found in PDB with those in SWISS-PROT (cf. A.103), GSDB (cf. A.38), or GenBank (cf. A.36).

Data modelling, data storage, and data acquisition. The schema of 3DBase is modelled using OPM. the underlying DBMS is the relational system Sybase. Two primary objects in 3DBase are "oExperiment" and "oMacroMolecule". These objects describe the experiment and the biologically active molecule, extending the current view found in PDB entries.

Data retrieval and query answering. 3DBase seems to be accessible through the 3DBrowser at several mirror sites all over the world. This provides a complex fixed-form query interface. There is no *ad-hoc* query interface.

There is no information whether the data can be downloaded or browsed by navigating through a hierarchy.

Size. No information available.

Literature on 3DBase:

[3DBa] E. Abola et al. 3DBase – a macromolecular structure database. Available at: `http://www.ornl.gov/TechResources/Human_Genome/publicat/96santa/informat/abola.html`

A.2 AAindex: The Amino Acid Index Database

Provider and access. AAindex, the amino acid index database is maintained at the Kyoto University, Kyoto/Japan. It can be accessed at: `http://www.genome.ad.jp/dbget/aaindex.html`

Contents. AAindex provides information about each of the 20 amino acids that occur in living organisms. Annotated data like biochemical properties and amino acid mutation matrices are attached. Links to other databases are provided.

Data modelling, data storage, and data acquisition. AAindex is a flat file database. It consists of two files, "aaindex1" containing the annotated amino acid index and "aaindex2" containing the amino acid mutation matrices. Both files are explicitly structured (aaindex2 only partly).

Data are acquired from the scientific literature.

Data retrieval and query answering. AAindex can only be queried via the DBGET/LinkDB (cf. Chapter 4.3.2) system. There are no query interfaces to directly access data from AAindex. There is also no capability to browse the data.

Both flat files of AAindex can be downloaded via FTP at: `ftp://ftp.genome.ad.jp/pub/db/genomenet/aaindex/`

Size. Currently both files of the AAindex database require approximately 500 KB (aaindex1: 341 KB; aaindex2: 158 KB).

Literature on AAindex:

[AAINa] S. Kawashima and M. Kanehisa. AAindex: Amino Acid Index Database. *Nucleic Acids Research*, 28(1):374, 2000.

[AAINb] `http://www.genome.ad.jp/dbget/aaindex.html`

A.3 AARSDB: The Aminoacyl-tRNA Synthetases Database

Provider and access. The Aminoacyl-tRNA Synthetases Database (AARSDB) is maintained at the Polish Academy of Science, Poznan, Poland. It can be accessed at: `http://rose.man.poznan.pl/aars/index.html`

Contents. AARSDB is a compilation of all Aaminoacyl-tRNA Synthetases which are available as separate entries or alignments of related proteins via the WWW. Links to external databases (primarily EMBL (cf. A.28)) are simple textual references.

Data modelling, data storage, and data acquisition. The AARSDB is a flat file database. Each entry is stored in a single file. The entries are structured explicitly using line types (search indexes). The format is based on the flat file format of EMBL (cf. A.28) and SWISS-PROT (cf. A.103). Data items are represented as HTML tables.

Data are acquired from the scientific literature.

Data retrieval and query answering. Data can only be browsed entering the data hierarchy provided by the AARSDB team. There are no query interfaces at all, and no possibilities to download the flat files.

Size. No information available.

Literature on AARSDB:

[AARSa] M. Szymanski and J. Braciszewski. Aminoacyl-tRNA Synthetases Database Y2K. *Nucleic Acids Research*, 28(1):326-328, 2000.

A.4 ALFRED: The Allele Frequency Database

Provider and access. The Allele Frequency Database (ALFRED) is maintained at the Yale University School of Medicine at New Haven, USA. It can be accessed at: `http://alfred.med.yale.edu/alfred/index.asp`

Contents. ALFRED has been designed to make allele frequency data on human population samples readily available to the scientific community and to link these polymorphism data to the molecular genetics-human genome databases. There are hypertext links to several other database.

Data modelling, data storage, and data acquisition. ALFRED is implemented using the relational DBMS Microsoft Access. The underlying relational schema of the database is documented at [ALFb].

Data are derived from the research community and the scientific literature.

Data retrieval and query answering. The ALFRED homepage provides several search facilities including a complex fixed-form query interface, and two browseable hierarchies. There is no *ad-hoc* query interface. The data are not released as a flat file and thus cannot be downloaded via FTP.

Size. No information available.

Literature on ALFRED:

[ALFa] K.-H. Cheung et al. ALFRED: an allele frequency database for diverse populations and DNA polymorphisms.*Nucleic Acids Research*, 28(1):361-363, 2000.

[ALFb] `http://alfred.med.yale.edu/alfred/table_list.asp`

A.5 aMAZE

Provider and access. The aMAZE database is under construction at the European Bioinformatics Institute (EBI). Some prototype features and the status of the project can be found at: `http://www.ebi.ac.uk/research/pfbp`.

Contents. The aim of the aMAZE project is to develop a database containing information on protein function, metabolic pathways and other biochemical pathways such as regulatory networks and signal transduction pathways. Links to sequence databases such as SWISS-PROT (cf. A.103), EMBL (cf. A.28), GenBank (cf. A.36), and 3D structure information of PDB (cf. A.75) will be provided.

Data modelling, data storage, and data acquisition. The data in aMAZE are represented using an object-oriented model. The commercial object DBMSs ObjectStore is used for data management. The model consists of three main types of objects [aMAZEb], representing:

- biochemical entities,
- biochemical activities, and
- pathways.

The data model was designed with special respect to manage schema evolutions. [aMAZEa] provides with more details on the data model.

Data are acquired from three other databases – KEGG (cf. A.52), SWISS-PROT (cf. A.103), and BRENDA (cf. A.11) – by parsing their contents. All parsed data have been checked for consistency.

Data retrieval and query answering. The external level of the aMAZE database is still under construction. It is expected that there will only be fixed-form query interfaces but no *ad-hoc* query interfaces. There will also be no flat file release, and probably no browsing capability.

Some tools for editing and creating pathways are available as prototypes at [aMAZEb].

Size. No information available.

Literature on aMAZE:

[aMAZEa] J. van Helden et al. From Molecular Activities and Processes to Biological Function. *Briefings in Bioinformatics*, 2(1):81-93, 2001.

[aMAZEb] `http://www.ebi.ac.uk/research/pfbp`

A.6 AMmtDB: A Metazoa Mitochondrial DNA Sequence Database

Provider and access. The database of multi-aligned *Metazoa* mitochondrial DNA sequences is maintained at CNR in Bari/Italy. It is accessible at: `http://bio-www.ba.cnr.it:8000/srs6`

Contents. AMmtDB is a database collecting the multi-aligned sequences of *Vertebrate* mitochondrial genes coding for proteins and tRNAs. The genes coding for proteins are multi-aligned based on the translated sequences and both the nucleotide and amino acid multi-alignments are provided. AMmtDB is interconnected with external databases through the Sequence Retrieval System (SRS, cf. Chapter 4.3.7).

Data modelling, data storage, and data acquisition. The database is organised into three main sections. Data are stored in flat files. The AMmtDB flat file format is defined at [AMmtDBb]. It consists of ten line types.

Data are mainly acquired from EMBL (cf. A.28) and GenBank (cf. A.36). This acquisition is done automatically by a tool called ”ACNUC”. Another source of data is the scientific literature.

Data retrieval and query answering. Data can only be retrieved via the Sequence Retrieval System (SRS, cf. Chapter 4.3.7). SRS integrates several databases and provides *ad-hoc* and fixed form query interfaces.

The flat files cannot be downloaded. There is no data classification to browse through.

Size. No information available.

Literature on AMmtDB:

[AMDBa] C. Lanave et al. Update of AMmtDB: a database of multi-aligned metazoa mitochondrial DNA sequences. nar , 28(1):153-154, 2000.

[AMDBb] `http://bio-www.ba.cnr.it:8000/Tutorials/AMmtDB/AMmtDBff.html`

A.7 ASDB: The Alternative Splicing Data Base

Provider and access. The Alternative Splicing Data Base (ASDB) is maintained at the National Energy Research Scientific Computing Center at Berkeley, USA. It can be accessed at: `http://devnull.lbl.gov:8888/alt/index.html`

Contents. ASDB consists of two divisions, ASDB(proteins), which contains amino acid sequences, and ASDB(nucleotides) with genomic sequences. It provides information about alternative splicing.

Entries in ASDB(proteins) are hypertext links to the according SWISS-PROT entries, entries in ASDB(nucleotides) are hypertext links to according GenBank entries. Some entries also provide hypertext links to other external databases such as EMBL (cf. A.28).

Data modelling, data storage, and data acquisition. Unfortunately, there is no information available about the underlying data model and the data management system. It seems that ASDB only manage links to the according data items in SWISS-PROT and GenBank.

Data are acquired from SWISS-PROT (cf. A.103) and GenBank (cf. A.36).

Data retrieval and query answering. ASDB can be searched through a collection of fixed-form query interfaces. Keywords can be searched against several search indexes. Two or more keywords can be combined with logical operators (only AND and OR). There is no *ad-hoc* query interface and no browsable hierarchy to retrieve data.

There is no possibility to download the data of ASDB.

Size. No information available.

Literature on ASDB:

[ASDBa] I. Dralyuk et al. ASDB: database of alternatively spliced genes. *Nucleic Acids Research*, 28(1):296-297, 2000.

[ASDBb] `http://devnull.lbl.gov:8888/alt/index.html`

A.8 AtDB

see TAIR (A.106 on page 229)

A.9 Axeldb: A *Xenopus laevis* Database

Provider and access. A *Xenopus laevis* Database (Axeldb) is maintained by the Deutsches Krebsforschungszentrum at Heidelberg/Germany and accessible at: `http://www.dkfz-heidelberg.de/abt0135/axeldb.htm`

Contents. Axeldb stores and integrates gene expression patterns and DNA sequences. Hypertext links to EMBL (cf. A.28) and PubMed (cf. A.90) are provided.

Data modelling, data storage, and data acquisition. Axeldb is implemented using the ACEDB software as DBMS. The database runs on an UNIX platform. The schema consists of 22 classes. The number of instances of one class ranges from 0 (no instances) to approximately 5 000.

Data are derived from the scientific community, some external databases, and the literature. Incoming data are originally in the form of ASCII flat files and are converted into the ace format files required by the ACEDB software. This is done using Perl and shell scripts.

Data retrieval and query answering. Axeldb can be browsed and searched in various ways:

All objects of Axeldb can be accessed by browsing through the classes of objects. Browsing can start based on the domains of one or more expression (fixed-form interface) or picture-oriented.

Several fixed-form query interfaces enable the selection of genes, clones, pictures, sequences etc. The database can also be searched using text keywords or via an *ad-hoc* query interface using the AQL.

BLAST [4] and FASTA [56] provide sequence similarity search against Axeldb.

Data can also be downloaded as a flat file via FTP.

Size. No information available.

Literature on Axeldb:

[Axela] N. Pollet et al. Axeldb: a Xenopus laevis database focusing on gene expression. *Nucleic Acids Research*, 28(1):139-140, 2000.

[Axelb] `http://www.dkfz-heidelberg.de/abt0135/axeldb.htm`

A.10 BMRB: BioMagResBank

Provider and access. The BioMagResBank (BMRB) is provided by the University of Wisconsin-Madison. It is accessible at: `http://www.bmrb.wisc.edu`.

Contents. BMRB provides data on three dimensional structures of peptides, proteins, and nucleic acids resulted from NMR experiments. It is cross-linked with the PDB (cf. A.75) and the NDB (cf. A.71).

Data modelling, data storage, and data acquisition. BMRB is internally managed by a relational DBMS. The data is converted into the so-called “Self-Defining Text Archive and Retrieval” (STAR) format [32] at the University of Wisconsin-Madison. The STAR format is used as data exchange format and format of the flat file release. Several tools are used to produce the STAR files (see [BMRBb]). A detailed description of the relational tables of the BMRB schema can be obtained at [BMRBc].

Data in BMRB are originally acquired from the literature and two external databases (PDB and NDB, see above). In addition, data submission from the research community is also accepted. Special data submission tools are provided to assist biochemical researchers in producing correct STAR files for data submission to BMRB [BMRBb].

Data retrieval and query answering. BMRB provides a fixed-form query interface where the user can query the database against one or more of eight different attributes. An interface to the FASTA [56] sequence comparison tool is also provided. The user can compare a DNA or protein sequence against the database [BMRBb]. There is no *ad-hoc* query interface.

In addition, the data is organised into a hierarchy that can be browsed via the so-called. The data can also be downloaded in flat files (in STAR format) via FTP at: `ftp://ftp.bmrb.wisc.edu`

Size. No information available.

Literature on BMRB:

[BMRBa] B. Seavey et al. A relational database for sequence-specific protein NMR data. *Journal of Biomolecular NMR*, 1:217-236, 1991.

[BMRBb] `http://www.bmrb.wisc.edu/`

[BMRBc] `http://www.bmrb.wisc.edu/pub/data/export_db/doc/index.html`

A.11 BRENDA, the Enzyme Database

Provider and access. BRENDA, the enzyme database is maintained at the University of Cologne, Germany. It can be accessed at: `http://www.brenda.uni-koeln.de/`

Contents. BRENDA is the main collection of enzyme functional data available to the scientific community. It provides several properties for each stored enzyme such as nomenclature, reaction-related annotation, structure-related annotation, and other pieces of information. See [BRENb] for details.

Hpertext links to external databases such as KEGG (cf. A.52) and SWISS-PROT (cf. A.103) and textual references to the literature are provided.

Data modelling, data storage, and data acquisition. Data in BRENDA are stored in the relational DBMS MySQL. There is no information available about the details of the underlying relational data model. For query answering, single data items are converted into flat files that are explicitly structured by line types (search indexes).

Data are acquired originally from the scientific literature. In addition, researchers can submit their data to BRANDA using a submission tool.

Data retrieval and query answering. BRENDA provides only fixed-form query interfaces. Three simple interfaces allow to query a keyword against a special search index. A further more complex interface provides the capability to express a query against two search index of the database combined with boolean operators. The queries are transformed into SQL statements and processed by the DBMS.

There is no *ad-hoc* query interface nor a browsable hierarchy. The data are also not released as a flat file for download.

Size. No information available.

Literature on BRENDA:

[BRENa] `http://www.brenda.uni-koeln.de/brenda/brenda_intro.html`

A.12 The CATH protein domains database

Provider and access. The CATH protein domains database (CATH) is maintained by the University of London, UK. It can be accessed at: `http://www.biochem.ucl.ac.uk/bsm/cath_new/`

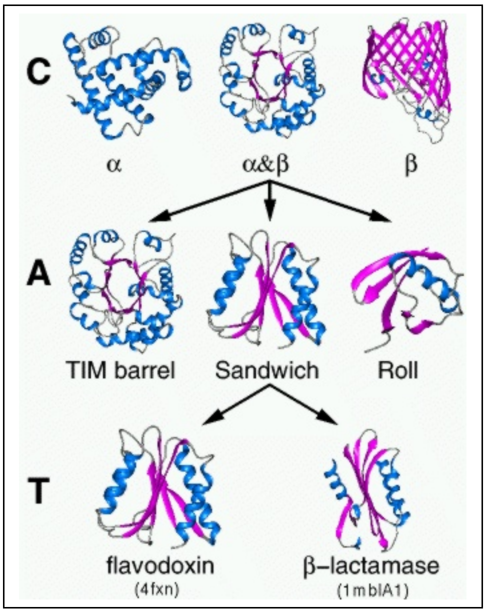


FIGURE A.1: Hierarchical of the CATH classification (taken from [CATHc]).

Contents. CATH provides a hierarchical classification of protein domains into evolutionary families and structural groupings.

The proteins in CATH are clustered at four major levels, Class (C), Architecture (A), Topology (T) and Homologous superfamily (H) (This is where the name CATH derives). The classification is carried out in seven steps including sequence comparison (S-level), assignment of domain boundaries for multidomain proteins, automatic assignment of classes, structure comparisons (H- and T-levels), assigning architectures, collecting data on individual structures, and finally assigning CATH numbers. Details can be found in [CATHa]. The hierarchical nature of the CATH classification is shown in Figure A.1 (top three levels C, A, T only).

In addition, the so-called CATH Protein Family Database (CATH-PFDB) which associates one-dimensional sequences with the three-dimensional homologous superfamilies and the CATH-DSH (Dictionary on Homologous Superfamilies) containing annotated multiple structural alignments are provided together with CATH.

There hypertext links to other databases such as PDB (cf. A.75), SWISS-PROT (cf. A.103) and several other external databases.

Data modelling, data storage, and data acquisition. There are currently no information available about the underlying data model and the data management system of CATH.

Data seem to be acquired from PDB (cf. A.75).

Data retrieval and query answering. There are several ways to retrieve data from CATH: First, the user can browse through the CATH hierarchy to retrieve single data items. Second, the CATH database can be searched via a fixed-form query interface by keywords or PDB identifiers. There is no *ad-hoc* query interface. Last, the entire contents of CATH can be downloaded via FTP as flat files at: `ftp://ftp.biochem.ucl.ac.uk/pub/cathdata/v2.0/`.

Size. All flat files of CATH together currently require about 4 MB.

Literature on CATH:

[CATHa] C. Orengo et al. CATH – a hierarchic classification of protein domain structures. *Structure*, 5(8):1093-1108, 1997.

[CATHb] F. Pearl et al. Assigning genomic sequences to CATH. *Nucleic Acids Research*, 28(1):277-282, 2000.

[CATHc] `http://www.biochem.ucl.ac.uk/bsm/cath_new/cath_info.html`

A.13 COG: The Database of Clusters of Orthologous Groups of proteins

Provider and access. The database of Clusters of Orthologous Groups of proteins (COG) is available at the home-page of the National Center for Biotechnology Information (NCBI) at: `http://www.ncbi.nlm.nih.gov/COG`.

Contents. The database is an attempt on phylogenetic classification of proteins encoded in complete genomes of bacteria, archaea and eucaryotes. Protein sequences are clustered by biochemical criteria into clusters of orthologous groups (COGs). The COGs are classified by 17 functional categories (see [COGa]).

Hypertext links are provided to GenBank (cf. A.36).

Data modelling, data storage, and data acquisition. The data are stored in flat files. Each COG is stored in an own file. The flat files consist of entries (representing proteins). Each entry is marked with ">" at the beginning and consists of a primary key followed by the amino acid sequence of the protein in the following lines. See Figure A.2 for a sample entry. Additional several documentation files contain e.g. information on each COG or on the abbreviations used as keywords are available.

New proteins are fitted into the COGs with a program called COGNITOR which is accompanied by the database. The original COGs have been identified on the basis of an all-against-all sequence comparison of the proteins encoded in complete sequenced genomes using the BLAST [4] program.

There is no information about curating available.

```
>AF1241
MKLDKSRKLYAEALNLMPGGVSSPVRAFKPHPFYTARGKGSKIYDVDGNAYIDYCMAYGPLVLGHANEVV
KNALAEQLERGWLYGTPIELEIEYAKLIQKYFPSMEMLRFVNTGSEATMAALRVARGFTGRDKIVKVEGS
FHGAHDAVLVKAGSGATTHGIPNSAGVPADFVKNTLQVPFNDIEALSEILEKNEVAALILEPVMGNSSLI
LPEKDYLKEVRKVTAENDVLLIFDEVITGFRVSMGGAQEYYGVKPDLTTLGKIAGGGLPIGIFGGRKEIM
ERVAPSGDVYQAGTFSGNPLSLTAGYATVKFMEENGVIEKVNSLTEKLVSGIADVLEDKKAECEVGSLAS
MFCIYFGPTPRNYAEALQLNKERFMEFFWRMLENGVFLPPSQYETCFVSFAHTEEDVEKTVEAVSESL
```

FIGURE A.2: Sample entry of COG.

Data retrieval and query answering. The data can be downloaded via FTP in a flat file at: `ftp://ncbi.nlm.nih.gov/pub/COG`

The COG WWW site contains the following data which can be browsed:

1. a list of all COGs organised by the (predicted) functional categories, hyper-linked to
2. individual COG pages: each of the COG pages shows the respective phylogenetic pattern, hyper-linked to
   - pictorial representations of BLAST [4] search outputs for each member of the COG (with hypertext links to GenBank and Entrez),
   - a multiple alignment of the COG members produced automatically by the ClustalW program,
   - a cluster dendrogram generated using BLAST [4] scores as the measure of similarity between proteins;
3. COGNITOR page were a protein sequence can be pasted, searched and assigned to a COG;
4. a phylogenetic pattern search tool;
5. a matrix of co-occurrence of genomes in COGs;

There are no *ad-hoc*, nor fixed-form query interfaces. The data files can be downloaded via FTP.

Size. The database consists of 2256 files, each representing a COG. The number of entries (proteins) in the files (COGs) differ from 3 to over 50. The individual files require between less than 1 KB and more than 150 KB. The file containing a collection of all COGs require (as zipped UNIX Tape Archive) currently approximately 15 000 KB.

Literature on COG:

[COGa] R. Tatusov et al. The COG database: a tool for genome-scalea analysis of protein functions and evolution. *Nucleic Acids Research*, 28(1):, 33-36, 2000.

[COGb] `http://www.ncbi.nlm.nih.gov/COG/COGhelp.html`

A.14 Colibri

Provider and access. Colibri is maintained by the Institut Pasteur in Paris/France and is publicly accessible at URL: `http://genolist.pasteur.fr/Colibri/`

Contents. Colibri integrates various aspects of the genomic information from *E. coli*. The database provides a complete data set of DNA and protein sequences, linked to the relevant annotations and functional assignments.

Hypertext links to GenBank (cf. A.36) and SWISS-PROT (cf. A.103) are provided.

Data modelling, data storage, and data acquisition. Colibri is implemented using the commercial relational DBMS Sybase. The schema is described comprehensively in [COLa]. New data are no longer acquired as the *E. coli* genome is fully sequenced.

Data of Colibri originate from the *E. coli* Genome Project and the EcoGene database (cf. A.26).

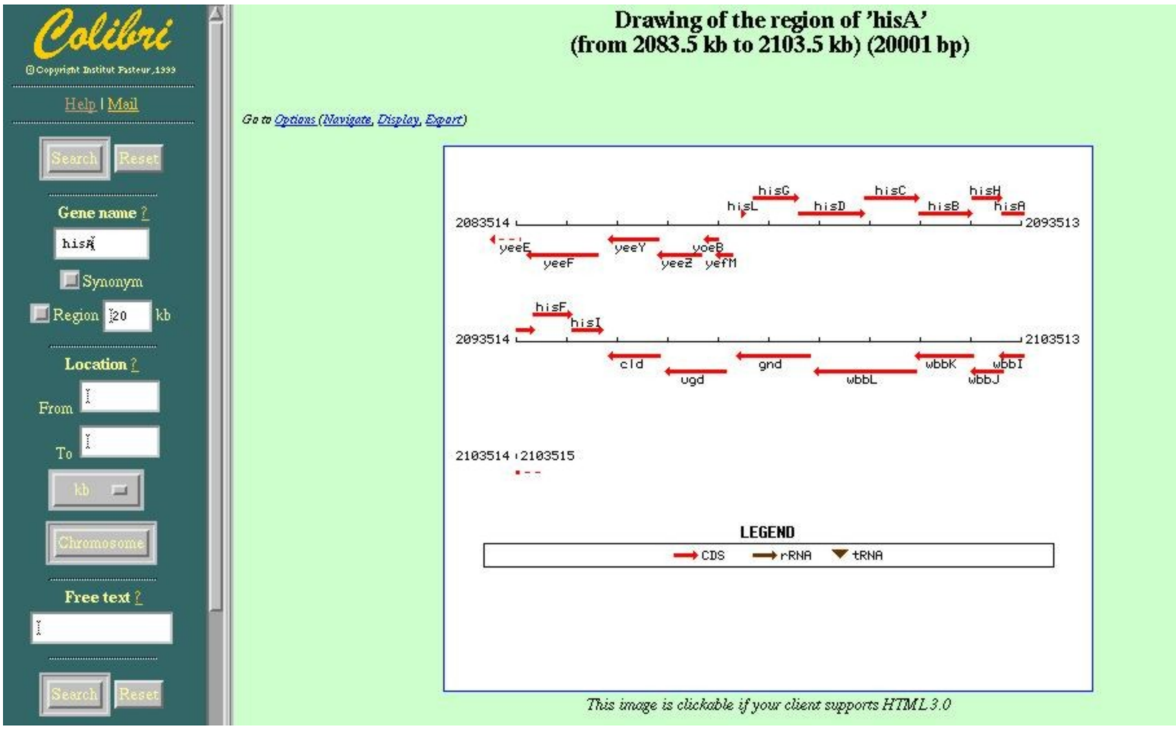


FIGURE A.3: Screenshot of the Colibri www-interface: The query interface is located on the left margin. A browsable map of the region of a sample gene is located in the center right.

Data retrieval and query answering. There are several fixed-form query interfaces, and an interface for text search. *Ad-hoc* queries can also be expressed in an home-made query language that tries to translate some SQL constructs into natural language (the query language allows constructs such as ”consists of”, ”like”, ”a part of”, etc.). The results are structured in HTML-tables and further analysis is supported by dedicated analysis tools.

In addition, a flat file distribution of the data contained in Colibri will be available soon. It will include a number of files (or series of files):

- The complete database in EMBL (cf. A.28) format (DNA sequence + associated features)
- The complete genome sequence in FASTA [56] format

- The protein-encoding DNA sequences in FASTA [56] format
- The protein sequences in FASTA [56] format

There are also capabilities to retrieve data by browsing e.g. through interactive maps (see Figure A.3).

Size. No information available.

Literature on Colibri:

[COLa] C.Medigue et al. Colibri: a functional data base for the Escherichia coli genome. *Microbiology and Molecular Biology Reviews*, 57(3):623-654, 1993.

[COLb] `http://genolist.pasteur.fr/Colibri/`

A.15 COMPEL

Provider and access. COMPEL is maintained by a part of the Siberian Branch of the Russian Academy of Science at Novosibirsk, Russia in collaboration with the Gesellschaft für Biotechnologische Forschung in Braunschweig, Germany. It is accessible at :`http://compel.bionet.nsc.ru`

Contents. COMPEL is a database on composite regulatory elements. It provides hypertext links to external databases such as TRANSFAC (cf. A.109) and TRRD (cf. A.112).

Data modelling, data storage, and data acquisition. COMPEL relies on a relational schema consisting of 16 tables. The data are managed by a relational DBMS. Further information about the COMPEL schema can be found in [COMPa].

Data are acquired probably from external databases and the literature.

Data retrieval and query answering. COMPEL can be searched through a fixed-form query interface. A keyword can be queried against several search indexes. There is no *ad-hoc* query interface. The data in COMPEL are also classified in a hierarchy. This hierarchy can be browsed to access single data items.

In addition, the database can be downloaded as flat file after registration (free!).

Size. No information available.

Literature on COMPEL:

[COMPa] O.Kel-Margoulis et al. COMPEL: a database on composite regulatory elements providing combinatorial transcriptional regulation. *Nucleic Acids Research*, 28(1):311-315, 2000.

[COMPb] `http://compel.bionet.nsc.ru/compel/description.html`

A.16 CSNDB: The Cell Signaling Networks Database

Provider and access. The cell signaling networks database (CSNDB) is maintained at the Japanese National Institute of Health Sciences and can be accessed at: `http://geo.nihs.go.jp/csndb/`

Contents. CSNDB provides data on cell signaling networks from various organisms. Its aim is to develop more realistic and quantitative models for biological phenomena. Links to the TRANSFAC (cf. A.109) and PubMed (cf. A.90) databases are implemented.

Data modelling, data storage, and data acquisition. The data are modelled and managed using the ACEDB software. The complete model can be browsed at [CSNb].

Data are acquired exclusively from the scientific literature.

Data retrieval and query answering. CSNDB can be queried via several different capabilities:

1. Fixed-form query interface: CSNDB provides a simple keyword search and a complex fixed-form query interface.

2. *Ad-hoc* query interface: CSNDB can be queried using the ACEDB query language.

In addition, the CSNDB providers classified within an hierarchy. This hierarchy can be browsed to retrieve single data items.

The ace-files can be downloaded via FTP at: `http://geo.nihs.go.jp/csndb/acedata/`

Size. No information available.

Literature on CSNDB:

[CSNa] T. Igarashi and T. Kaminuma. Development of a cell signaling networks database. In *Pacific Symposium on Biocomputing*, 187-197, 1997.

[CSNb] `http://geo.nihs.go.jp/cgi-bin/model/wwwace`

A.17 CyanoBase

Provider and access. CyanoBase is build and maintained by the Kazusa DNA Research Institute at Chiba/Japan and freely accessible at: `http://www.kazusa.or.jp/cyano/`

Contents. CyanoBase provides an online resource for access to data on genomic information about the cyanobacterium *Synechocystis sp.* The database contains annotations for each protein-coding gene.

Every entry in the database provides a hypertext link to an according GenBank (cf. A.36) entry via the integration system DBGET/LinkDB at Genome Net (cf. Chapter 4.3.2).

Data modelling, data storage, and data acquisition. The database is build upon a relational DBMS (Sybase). There is no information available about the underlying model. Map interfaces and presentation of the annotation to each gene were implemented in the home-made JMGD system. The JMGD software package is available via FTP and includes the Java source code and a set of Perl scripts. A file of the database in ACEDB format is prepared separately.

Data are acquired from the entire nucleotide sequence databases GenBank (cf. A.36), EMBL (cf. A.28), and DDBJ (cf. A.22) and private submissions from the community.

Data retrieval and query answering. Data can be retrieved indirectly by browsing a clickable map of the genome of *Synechocystis sp.* (eighter a Java or a GIF map). When a given position of the map is clicked, a local map covering the corresponding 90 kb area appears. A second indirect way to retrieve data from CyanoBase is by a gene category list. This list is a table of gene classification, presented in an hierarchical manner, with a link to the annotation to each gene.

An interface provides simple keyword search. Similarity search against CanoBase can also be performed. Additional, there are several analysis tools with the database.

Data can also be downloaded in several flat files via FTP. Special formats like FASTA [56] format or Microsoft EXCEL spread-sheets are provided. One flat file in ACEDB format is available.

There is no *ad-hoc* query interface.

Size. Data are partitioned in several flat files (e.g. one file contains only the nucleotide sequences, and another contains a table of potential protein coding regions), which sizes currently differ between approximately 70 and 1100 KB.

The ACEDB file of CyanoBase requires (as UNIX TAR file) about 21 MB.

Literature on CyanoBase:

[CYBa] Y. Nakamura, T. Kaneko, and S. Tabata. CyanoBase, the genome database for Synechocystis sp. strain PCC6803: status for the year 2000. *Nucleic Acids Research*, 28(1):72, 2000.

[CYBb] `http://www.kazusa.or.jp/cyano/`

A.18 DAtA: Database of *A. thaliana* Annotation

Provider and access. The Database of *A. thaliana* Annotation (DAtA) is maintained at Stanford DNA Sequencing and technology Center at Palo Alto/USA and is accessible at: `http://luggagefast.Stanford.EDU/group/arabprotein/`.

Contents. DAtA was created to enable easy access to and analysis of all the *Arabidopsis* genome project annotation. DAtA also contains protein motifs and protein similarities.

The database provides hypertext links to other databases, such as MIPS (cf. A.63) and Genbank (cf. A.36) via the integrated retrieval system Entrez at NCBI (cf. Chapter 4.3.4).

Data modelling, data storage, and data acquisition. The data are stored using the relational DBMS MySQL. Information about the schema are not available.

Data are primarily acquired from GenBank (cf. A.36). If no protein annotation is present in the acquired GenBankrecord, an automated annotation process is performed on the DNA sequence to generate protein predictions and other information.

Data retrieval and query answering. DAtA provides a fixed-form query interface. The database can be searched by several fields such as protein or gene name, or chromosome of the according gene, motif field, and BLAST [4] similarity.

Data are presented as shown in Figure A.4.

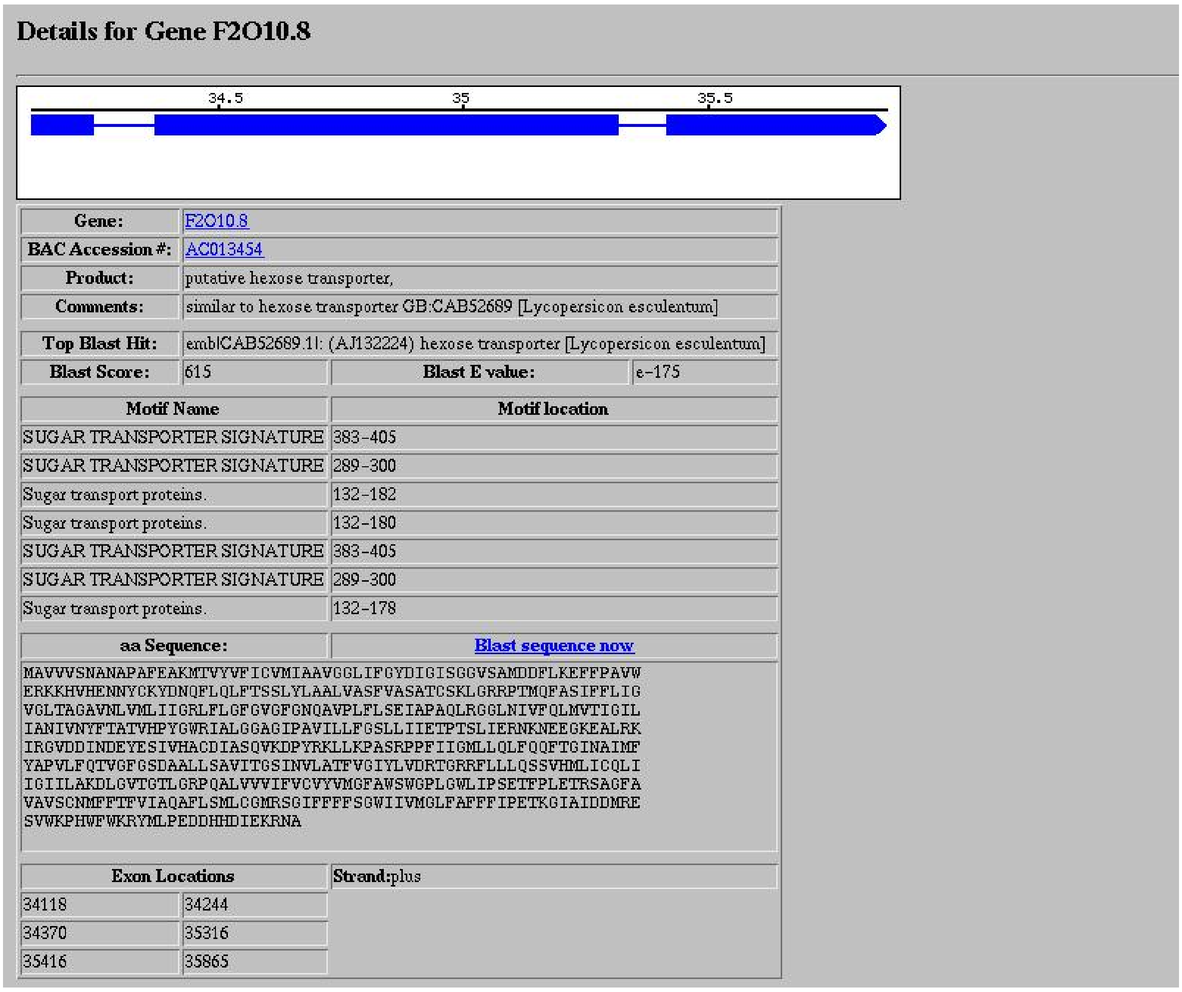
The database can also be browsed by protein names or motif names. The user can choose the protein/motif of interest out of a list of all available data.

The data of DAtA can be downloaded via FTP in FASTA [56] format for UNIX, PC and Macintosh platforms.

Ad-hoc queries cannot be performed against DAtA.

Size. DAtA currently stores about 115 MB of data. The database is routinely updated to include new GenBank submissions for *Arabidopsis* genomic sequences.

Literature on DAtA:



Details for Gene F2O10.8

Gene:	F2O10.8		
BAC Accession #:	AC013454		
Product:	putative hexose transporter,		
Comments:	similar to hexose transporter GB:CAB52689 [Lycopersicon esculentum]		
Top Blast Hit:	emb|CAB52689.1|: (AJ132224) hexose transporter [Lycopersicon esculentum]		
Blast Score:	615	**Blast E value:**	e-175

Motif Name	**Motif location**
SUGAR TRANSPORTER SIGNATURE	383-405
SUGAR TRANSPORTER SIGNATURE	289-300
Sugar transport proteins.	132-182
Sugar transport proteins.	132-180
SUGAR TRANSPORTER SIGNATURE	383-405
SUGAR TRANSPORTER SIGNATURE	289-300
Sugar transport proteins.	132-178

aa Sequence: **Blast sequence now**

```
MAVVVSNANAPAFEAKMTVYVFICVMIAAVGGLIFGYDIGISGGVSAMDDFLKEFFPAVW
ERKKHVHENNYCKYDNQFLQLFTSSLYLAALVASFVASATCSKLGRRPTMQFASIFFLIG
VGLTAGAVNLVMLIIGRLFLGFGVGFGNQAVPLFLSEIAPAQLRGGLNIVFQLMVTIGIL
IANIVNYFTATVHPYGWRIALGGAGIPAVILLFGSLLIIETPTSLIERNKNEEGKEALRK
IRGVDDINDEYESIVHACDIASQVKDPYRKLLKPASRPPFIIGMLLQLFQQFTGINAIMF
YAPVLFQTVGFGSDAALLSAVITGSINVLATFVGIYLVDRTGRRFLLLQSSVHMLICQLI
IGIILAKDLGVTGTLGRPQALVVVIFVCVYVMGFAWSWGPLGWLIPSETFPLETRSAGFA
VAVSCNMFFTFVIAQAFLSMLCGMRSGIFFFFSGWIIVMGLFAFFFIPETKGIAIDDMRE
SVWKPHWFWKRYMLPEDDHHDIEKRNA
```

Exon Locations		**Strand:**plus
34118	34244	
34370	35316	
35416	35865	

FIGURE A.4: Representation of the Information of a Sample Entry in DAtA

[DAtAa] C. Palm, N. Federspiel, and R. Davis. DAtA: database of Arabidopsis thaliana annotation. *Nucleic Acids Research*, 28(1):102-103, 2000.

[DAtAb] `http://luggagefast.Stanford.EDU/group/arabprotein/`

A.19 DBcat

Provider and access. DBcat maintained and curated at Infobiogen and is accessible at: `http://www.infobiogen.fr/services/dbcat`.

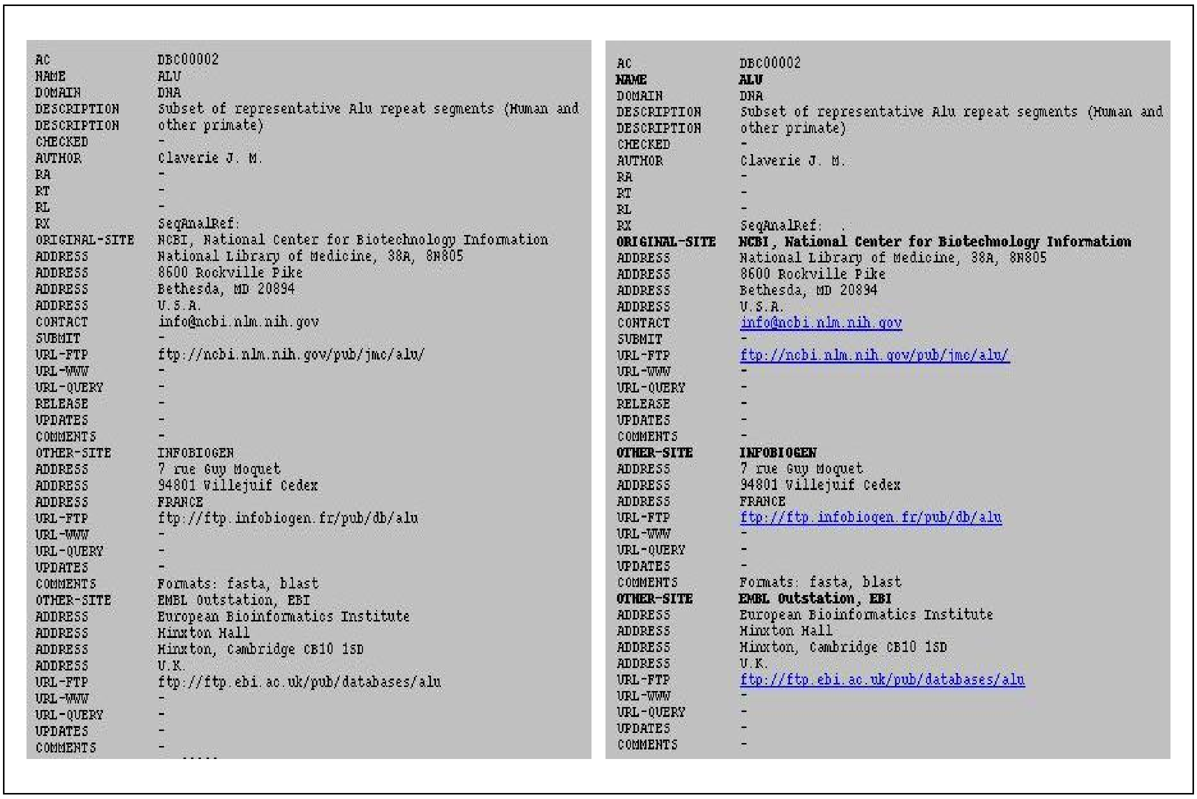
Contents. DBcat is a comprehensive catalog of molecular biology databases. Both kinds of links to other databases are provided:

- Textual references: in flat-files there are texts telling the URL of the provider's homepage, the

URL with the location where to download flat-files (if existing), and the URL of the query interfaces (if existing) of the according databases.

- Hypertext links: the result of querying DBcat by using the query by name interface at Infobiogen is represented in a browsable HTML-format (see Figure A.5, right). This format represents the URLs mentioned above as hypertext links.

Data modelling, data storage, and data acquisition. DBcat is implemented as one flat file. All entries are structured explicitly. The entry structure consists of a list of fields and their associated values (e.g. the "NAME" field describes the name of the database). The database entries are grouped into eight classes according to the application domain (e.g. DNA, RNA, Protein, etc.). The "DOMAIN" field of each entry indicates which classes the entry belongs to. It is remarkable that one entry may belong to more than one application domain. In addition to the flat file containing the whole DBcat data there are also flat files containing only the entries of one application domain. See Figure A.5 for a sample entry of DBcat in flat file format and in HTML format, as the result of a web-based query.



```
AC              DBC00002
NAME            ALU
DOMAIN          DNA
DESCRIPTION     Subset of representative Alu repeat segments (Human and
DESCRIPTION     other primate)
CHECKED         -
AUTHOR          Claverie J. M.
RA              -
RT              -
RL              -
RX              SeqAnalRef:
ORIGINAL-SITE   NCBI, National Center for Biotechnology Information
ADDRESS         National Library of Medicine, 38A, 8N805
ADDRESS         8600 Rockville Pike
ADDRESS         Bethesda, MD 20894
ADDRESS         U.S.A.
CONTACT         info@ncbi.nlm.nih.gov
SUBMIT          -
URL-FTP         ftp://ncbi.nlm.nih.gov/pub/jmc/alu/
URL-WWW         -
URL-QUERY       -
RELEASE         -
UPDATES         -
COMMENTS        -
OTHER-SITE      INFOBIOGEN
ADDRESS         7 rue Guy Moquet
ADDRESS         94801 Villejuif Cedex
ADDRESS         FRANCE
URL-FTP         ftp://ftp.infobiogen.fr/pub/db/alu
URL-WWW         -
URL-QUERY       -
UPDATES         -
COMMENTS        Formats: fasta, blast
OTHER-SITE      EMBL Outstation, EBI
ADDRESS         European Bioinformatics Institute
ADDRESS         Hinxton Hall
ADDRESS         Hinxton, Cambridge CB10 1SD
ADDRESS         U.K.
URL-FTP         ftp://ftp.ebi.ac.uk/pub/databases/alu
URL-WWW         -
URL-QUERY       -
UPDATES         -
COMMENTS        -
                  .....
```

```
AC              DBC00002
NAME            ALU
DOMAIN          DNA
DESCRIPTION     Subset of representative Alu repeat segments (Human and
DESCRIPTION     other primate)
CHECKED         -
AUTHOR          Claverie J. M.
RA              -
RT              -
RL              -
RX              SeqAnalRef: .
ORIGINAL-SITE   NCBI, National Center for Biotechnology Information
ADDRESS         National Library of Medicine, 38A, 8N805
ADDRESS         8600 Rockville Pike
ADDRESS         Bethesda, MD 20894
ADDRESS         U.S.A.
CONTACT         info@ncbi.nlm.nih.gov
SUBMIT          -
URL-FTP         ftp://ncbi.nlm.nih.gov/pub/jmc/alu/
URL-WWW         -
URL-QUERY       -
RELEASE         -
UPDATES         -
COMMENTS        -
OTHER-SITE      INFOBIOGEN
ADDRESS         7 rue Guy Moquet
ADDRESS         94801 Villejuif Cedex
ADDRESS         FRANCE
URL-FTP         ftp://ftp.infobiogen.fr/pub/db/alu
URL-WWW         -
URL-QUERY       -
UPDATES         -
COMMENTS        Formats: fasta, blast
OTHER-SITE      EMBL Outstation, EBI
ADDRESS         European Bioinformatics Institute
ADDRESS         Hinxton Hall
ADDRESS         Hinxton, Cambridge CB10 1SD
ADDRESS         U.K.
URL-FTP         ftp://ftp.ebi.ac.uk/pub/databases/alu
URL-WWW         -
URL-QUERY       -
UPDATES         -
COMMENTS        -
```

FIGURE A.5: Sample entry of DBcat in flat-file format (left) and in "result of query" format (right) (available at [DBCATb])

Data for DBcat are acquired from the literature, another catalog of molecular biology analysis tools called BioCatalog (provided at EBI), and private submissions via a Web form. New databases are also searched in the Web by search engines or biology-oriented web-sites. Providers of databases that are added to DBcat can validate the entry of their databases. After that each validated entry in DBcat is marked as "CHECKED".

Data retrieval and query answering. Data of DBcat are accessible in three ways:

- The flat-files of DBcat can be downloaded via FTP at: `ftp://ftp.infobiogen.fr/pub/db/dbcat`
- A web interface is available at the Infobiogen homepage [DBCATb] with a simple query by name interface and a possibility to browse through a classification by contents of the databases.
- Data of DBcat can also be retrieved by the Sequence Retrieval System of EBI (SRS, cf. Chapter 4.3.7), e.g. at: `http://www.infobiogen.fr/srs/`

There is no *ad-hoc* query interface at Infobiogen.

Size. In the end of October 2000, DBcat consists of 513 databases classified into eight classes. In November 2000, the flat file requires 594 KB. The sizes of the eight flat files representing the eight contents classes are between 19 KB (division "Structure") and 109 KB (division "Miscellaneous").

Literature on DBcat:

[DBCATa] C. Discala et al. DBcat: a catalog of 500 biological databases. *Nucleic Acids Research*, 28(1):8-9, 2000.

[DBCATb] `http://www.infobiogen.fr/services/dbcat`

A.20 dbEST

dbEST is a division of GenBank (cf. A.36) providing annotated EST sequence data.

A.21 dbSNP

Provider and access. dbSNP is maintained at the National Center for Biotechnology Information (NCBI) at Bethesda, USA. It can be accessed at: `http://www.ncbi.nlm.nih.gov/SNP/`

Contents. dbSNP is a database on single nucleotide polymorphisms (SNPs) to address the large-scale sampling designs required by association studies, gene mapping, and evolutionary biology. It is cross-linked with other databases at the NCBI such as GenBank (cf. A.36) and PubMed (cf. A.90).

Data modelling, data storage, and data acquisition. dbSNP is implemented using the relational DBMS Sybase. The underlying entity-relationships model is documented at [SNPb].

Data are acquired from the other NCBI databases (primarily GenBank, cf. A.36) and the research community. A submission tool manages the submissions of the research community.

Data retrieval and query answering. dbSNP can be searched via several complex fixed-form query interfaces. There is no *ad-hoc* query interface (although two fixed-form query interfaces are termed "free-form query interfaces").

The queries are most likely executed by Entrez (cf. Chapter 4.3.4), the database integration tool and querying system of NCBI.

There is no hierarchy to browse through for data retrieval.

In addition, the entire database is distributed in several flat files. The entries in these files are explicitly structured using line types. The files can be downloaded via FTP at: `ftp://ncbi.nlm.nih.gov/snp/`

Size. No information available.

Literature on dbSNP:

[SNPa] J. Ponomarenko. dbSNP: a database of single nucleotide polymorphisms. *Nucleic Acids Research*, 28(1):352-355, 2000.

[SNPb] `http://www.ncbi.nlm.nih.gov/SNP/dbSNP_work000401_gif_frame.htm`

A.22 DDBJ: The DNA Data Bank of Japan

Provider and access. The DNA Data Bank of Japan (DDBJ) is maintained at the National Institute of Genetics (NIG) in Mishima, Japan. It can be accessed at: `http://www.ddbj.nig.ac.jp/`

Contents. DDBJ provides all publicly available nucleotide sequences in tight collaboration with GenBank (cf. A.36) and EMBL (cf. A.28). Hypertext links to several external databases are provided.

Data modelling, data storage, and data acquisition. DDBJ is implemented using the relational DBMS Sybase. The underlying relational schema is discussed in [DDBJb].

Data are acquired primarily from the research community and the collaborating databases (see above). Specialised tools can be used to submit data to DDBJ.

Note: The three collaborating databases (DDBJ, EMBL, and GenBank) have defined a feature table that determines which information is added to the annotated sequences in all databases. In fact, the structure of the flat file release of each database is slightly different in syntax and semantics, but an entry in one of the collaborating databases contains exactly the same information as the corresponding entries in the other collaborating databases.

Data retrieval and query answering. DDBJ can be accessed through several fixed-form query interfaces. There is no *ad-hoc* query interface, nor a classification of the data that can be browsed. A local Sequence Retrieval System (SRS, cf. 4.3.7) server at NIG provides additional retrieval capabilities.

In addition, the entire data are released in several flat files. These files are available for download via FTP at: `ftp://ftp.ddbj.nig.ac.jp/`

Size. Detailed information is available at [DDBJc].

Literature on DDBJ:

[DDBJa] Y. Tateno et al. DNA Data Bank of Japan (DDBJ) in collaboration with mass sequencing teams. *Nucleic Acids Research*, 28(1):24-26, 2000.

[DDBJb] H. Kitakami et al. Building and search system for a large-scale DNA database. In: Schulze-Kremer S. (editor): *Advances in Molecular Bioinformatics*, IOS Press, 1994, 123-138

[DDBJc] `http://www.ddbj.nig.ac.jp/ddbjnew/statistics-e.html`

A.23 DIP: The Database of Interacting Proteins

Provider and access. The Database of Interacting Proteins (DIP) is maintained at UCLA, Los Angeles,USA. It is accessible at: `http://dip.doe-mbi.ucla.edu/`

Contents. DIP is a database that documents experimentally determined protein-protein interactions. It provides hypertext links to several other databases.

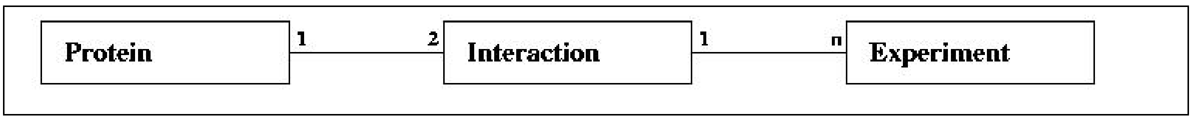


FIGURE A.6: Basic schema of DIP

Data modelling, data storage, and data acquisition. Originally, DIP was a flat file database. But to handle effectively the growing body of data, the database has been re-implemented. Now the data are stored in the relational DBMS MySQL. The schema of the database consists of three linked tables as shown in Figure A.6. The “Protein” table contains information about a protein entry and links (textual references) to sequence databases. The “Interaction” table describes information about a protein interaction. Each interaction is based on two participating proteins. The “Experiment” table describes details about experiments used to detect the interactions. One experiment can be determined by several experiments.

See [DIPa] for further information.

New data are currently only acquired from the scientific literature. New data items are added manually.

Data retrieval and query answering. DIP can be search via a fixed-form query interface. The search can be performed against the entire database (keyword search) or against several fields of DIP. Additional the data can be downloaded in flat files. A data file in a special format and a file in FASTA format are provided.

There is no *ad-hoc* query interface. There is also no hierarchy that can be browsed for indirect data retrieval.

Size. Currently the complete database as a file requires about 4.2 MB.

Literature on :

[DIPa] I. Xenarios et al. DIP: the database of interacting proteins. *Nucleic Acids Research*, 28(1):289-291, 2000.

[DIPb] `http://dip.doe-mbi.ucla.edu/`

A.24 DSMP: The Database of Structural Motifs in proteins

Provider and access. The database of structural motifs in proteins (DSMP) is available at: `http://www.cdfd.org.in/dsmp.html`.

Contents. DSMP contains data on structural motifs in proteins such as data relevant to helices, strands, and turns. The data were produced using the PROMOTIF program (developed by Hutchinson and Thornton, 1996) out of the 10 213 proteins in release 89 of the PDB (cf. A.75). Data are also available for a representative data set of 1 028 proteins. These data were also produced as mentioned above.

Data modelling, data storage, and data acquisition. The results of PROMOTIF are not available in web-based form so the structural motif data were converted into flat file format.

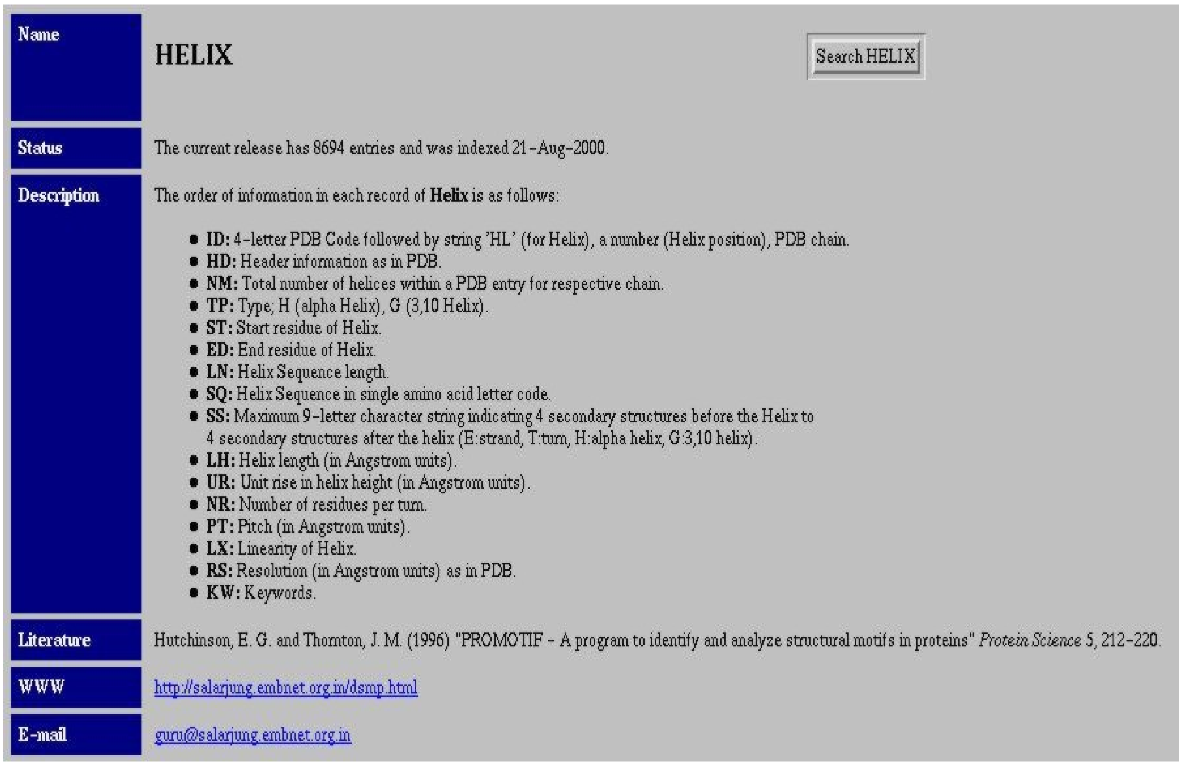
The data are organised in several flat files, each consisting entries with the same structural motif. E.g. in the "HELIX" file there are only entries providing a helix. Figure A.8 shows an information page about the HELIX section of DSMP generated by SRS (cf. Chapter 4.3.7). Each entry is explicitly structured using line types. A sample entry (of the HELIX section) of DSMP is given in Figure A.7.

```
ID   1a6dHL1A
HD   CHAPERONIN
NM   21
TP   H
ST   20
ED   40
LN   21
SQ   GKNAQRNNIEAAKAIADAVRT
SS      EHTTEE
LH   33.30
UR   1.58
NR   3.85
PT   6.07
LX   16.3
RS   2.6
KW   THERMOPLASMA ACIDOPHILUM, GROUP II CHAPERONIN, CCT, TRIC,
KW   2 PROTEIN FOLDING, ATPASE
//
```

FIGURE A.7: A sample entry of DSMP.

It is accounted that the structure of the entries may be extended in the future by adding new fields and new data.

Data are acquired from the PDB (cf. A.75) as mentioned above.



Name	HELIX [Search HELIX]
Status	The current release has 8694 entries and was indexed 21-Aug-2000.
Description	The order of information in each record of **Helix** is as follows: • **ID:** 4-letter PDB Code followed by string 'HL' (for Helix), a number (Helix position), PDB chain. • **HD:** Header information as in PDB. • **NM:** Total number of helices within a PDB entry for respective chain. • **TP:** Type; H (alpha Helix), G (3,10 Helix). • **ST:** Start residue of Helix. • **ED:** End residue of Helix. • **LN:** Helix Sequence length. • **SQ:** Helix Sequence in single amino acid letter code. • **SS:** Maximum 9-letter character string indicating 4 secondary structures before the Helix to 4 secondary structures after the helix (E:strand, T:turn, H:alpha helix, G:3,10 helix). • **LH:** Helix length (in Angstrom units). • **UR:** Unit rise in helix height (in Angstrom units). • **NR:** Number of residues per turn. • **PT:** Pitch (in Angstrom units). • **LX:** Linearity of Helix. • **RS:** Resolution (in Angstrom units) as in PDB. • **KW:** Keywords.
Literature	Hutchinson, E. G. and Thornton, J. M. (1996) "PROMOTIF - A program to identify and analyze structural motifs in proteins" *Protein Science* 5, 212-220.
WWW	http://salarjung.embnet.org.in/dsmp.html
E-mail	guru@salarjung.embnet.org.in

FIGURE A.8: Information about the HELIX section of DSMP.

Data retrieval and query answering. Data can only be retrieved using the Sequence Retrieval System (SRS, cf. Chapter 4.3.7).

There is a local SRS server that can be accessed via several entry-points at the DSMP home-page. The local SRS server can be used to query the database.

Size. Currently (as of Winter 2000) there are approximately 150 000 entries stored in 13 different flat files ([DSMPb]). DSMP is proposed to be updated periodically as new structures become available.

Literature on DSMP:

[DSMPa] C. Harger et al. Database of structural motifs in proteins. *Bioinformatics*,16(4):372-375, 2000.

[DSMPb] `http://www.cdfd.org.in/dsmp.html`

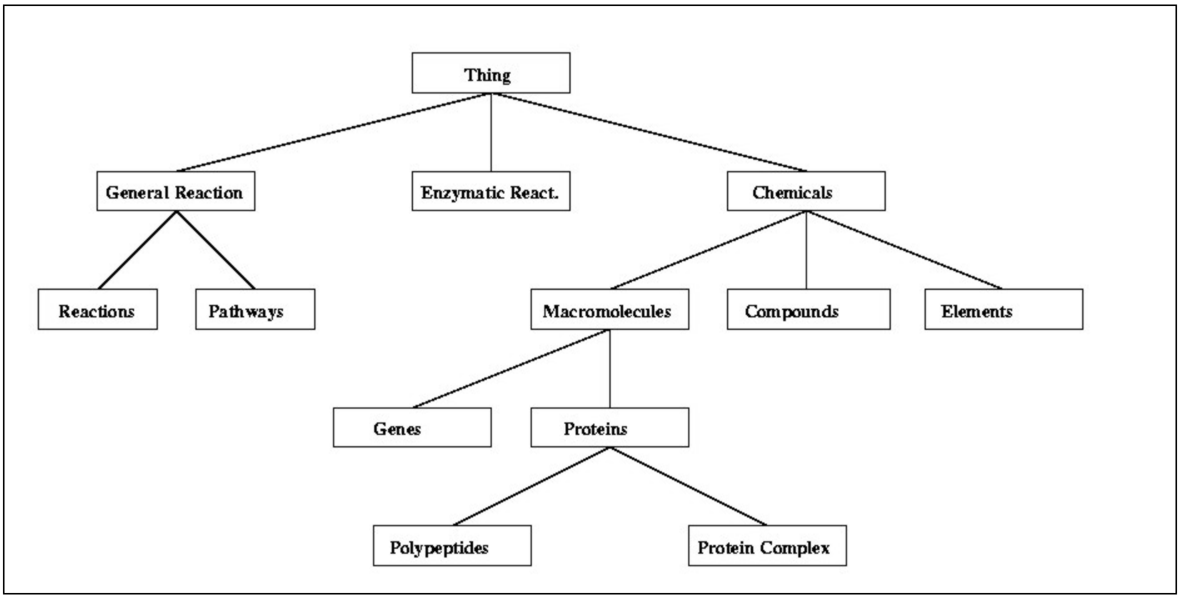


FIGURE A.9: Basic class hierarchy of the EcoCyc/MetaCyc schema (according to [ECOCb])

A.25 The EcoCyc and MetaCyc Databases

Provider and access. EcoCyc and MetaCyc databases are maintained at the Artificial Intelligence Center, SRI International, (California/USA) and are available at:
`http://ecocyc.pangeasystems.com/ecocyc/`

Contents. EcoCyc describes the metabolic and signal-transduction pathways of *E. coli*, its enzymes and its transport proteins. MetaCyc is a metabolic pathway database that describes pathways and enzymes of many different organisms, with a microbial focus.

Both databases overlap each others for they both contain all pathways of *E.coli* small-molecule metabolism.

Data modelling, data storage, and data acquisition. Both the EcoCyc and the MetaCyc data are stored within a frame knowledge representation system (FRS) called Ocelot. Ocelot is developed by the Ecocyc/MetaCyc group at SRI International. FRSs use an object model. Both databases are build upon the same schema. The basic class hierarchy of this schema is shown in Figure A.9.

Every class has a set of attributes describing their properties.

Pathway data are represented using a predecessor list (cf. Chapter 5 on this approach to model metabolic pathways).

EcoCyc is implemented in Common Lisp with a graphical interface toolkit called Common Lisp Interface Manager (CLIM). Several consistency constraints in the databases ensure the correctness of the existing data as well as of new entries.

Data were originally acquired mainly from EcoGene (cf. A.26), ENZYME (cf. A.30), several other databases, and the scientific literature.

New data seem to be acquired by the scientific literature, but there are no explicit information on the sources of new data.

Data retrieval and query answering. The user must be registered before accessing data from EcoCyc/MetaCyc (see [ECOCc]). Academic users are free of charge.

The data can be accessed through the web interface at the EcoCyc/MetaCyc homepage ([ECOCc]) or can be obtained via download as a software package that bundles together the GUI and the databases (for a local installation). This package only runs on a Sun workstation.

The web interface at [ECOCc] provides an integrated GUI with several analysis tools and (fixed-form) query interfaces. There is also the possibility to express *ad-hoc* queries in an "home-made" declarative query language. The results of a query can be saved in a history list. A comprehensive list of all fixed-form queries supported by the GUI of the two databases can be obtained from [ECOCb].

The user can also navigate following several hypertext links to access data (for example a browsable map of the *E. coli* genome is provided, or from a pathway drawing the user can navigate to a super-pathway that contains it, or *vice versa*.)

Size. EcoCyc consists of approximately 10 000 objects, MetaCyc consists of approximately 6 500 objects. Detailed information on the current size of each class (number of instances) can be obtained from [ECOCa].

Literature on EcoCyc and MetaCyc:

[ECOCa] P. Karp et al. The EcoCyc and MetaCyc Databases. *Nucleic Acids Research*, 28(1):56-59, 2000.

[ECOCb] P. Karp et al. EcoCyc: Encyclopedia of E.coli Genes and Metabolism. *Nucleic Acids Research*, 25(1):43-51, 1997.

[ECOCc] `http://ecocyc.pangeasystems.com/ecocyc/`

A.26 EcoGene: A Genome Sequence Database for *E.coli*

Provider and access. EcoGene is maintained at the University of Miami School of Medicine, Department of Biochemistry and Molecular Biology, Miami/USA and publicly available at: `http://bmb.med.miami.edu/EcoGene/EcoWeb`

Contents. EcoGene contains gene and protein sequences of *E. coli*. The original data were obtained from GenBank (cf. A.36) using several tools. Nevertheless, annotation of the sequences in EcoGene differs from GenBank annotation (see below). Thus, EcoGene provides an alternative view on the *E. coli* gene and protein annotation, which seemingly is useful for the design of other *E. coli* research and database projects. E.g. SWISS-PROT (cf. A.103), EcoCyc (cf. A.25), and Colibri (cf. A.14) are populated by (among others) the EcoGene data or utilise special features of EcoGene.

EcoGene provides more than 12 000 cross-references (hypertext links) to literature citations and abstracts

Data modelling, data storage, and data acquisition. EcoGene seems to have a relational data model. But all data are stored in flat files even the tables.

EcoGene flat files contain the following data sets:

- files storing the tables of the "data model"
- files storing the sequences of the genes and proteins (files containing all sequences are as well available as files containing only one individual sequence)
- a file containing a graphic representation of the gene map in PDF format

The gene map is converted out of the ECMAIN and ECMAP tables using the so called Print map C program. The list of tables can be viewed and browsed at: `http://bmb.med.miami.edu/EcoGene/EcoWeb/CESSPages/FILES/Tables/`.

New data are no longer acquired (the *E. coli* genome is fully sequenced).

Data retrieval and query answering. At [ECOGb] there is a simple text search interface. The results are represented via the so called GenePage as shown in Figure A.10. The GenePage provides several information and (among others) the following links to special datasets for each gene:

- Bibliography: bibliographic references according to the entry
- DNA Sequence: no-structured (raw ASCII) nucleotide sequence of the entry
- Protein Sequence: no-structured (raw ASCII) amino acid sequence of the product of the entry
- several hypertext-links to other databases (e.g. Colibri, SWISS-PROT, etc.)

No *ad-hoc* or free-form query interfaces are provided.

All datasets mentioned above can be retrieved by download via FTP, such as the PDF file with gene maps, gene sequences, tables (for each gene the gene tables tell the position of the gene on the genome, i.e. the start and end of the gene), etc.

Size. No information available.

Literature on EcoGene:

[ECOGa] K. Rudd. EcoGene: a Genome Sequence Database for E. coli K-12. *Nucleic Acids Research*, 28(1):60-64, 2000.

[ECOGb] `http://bmb.med.miami.edu/EcoGene/EcoWeb`

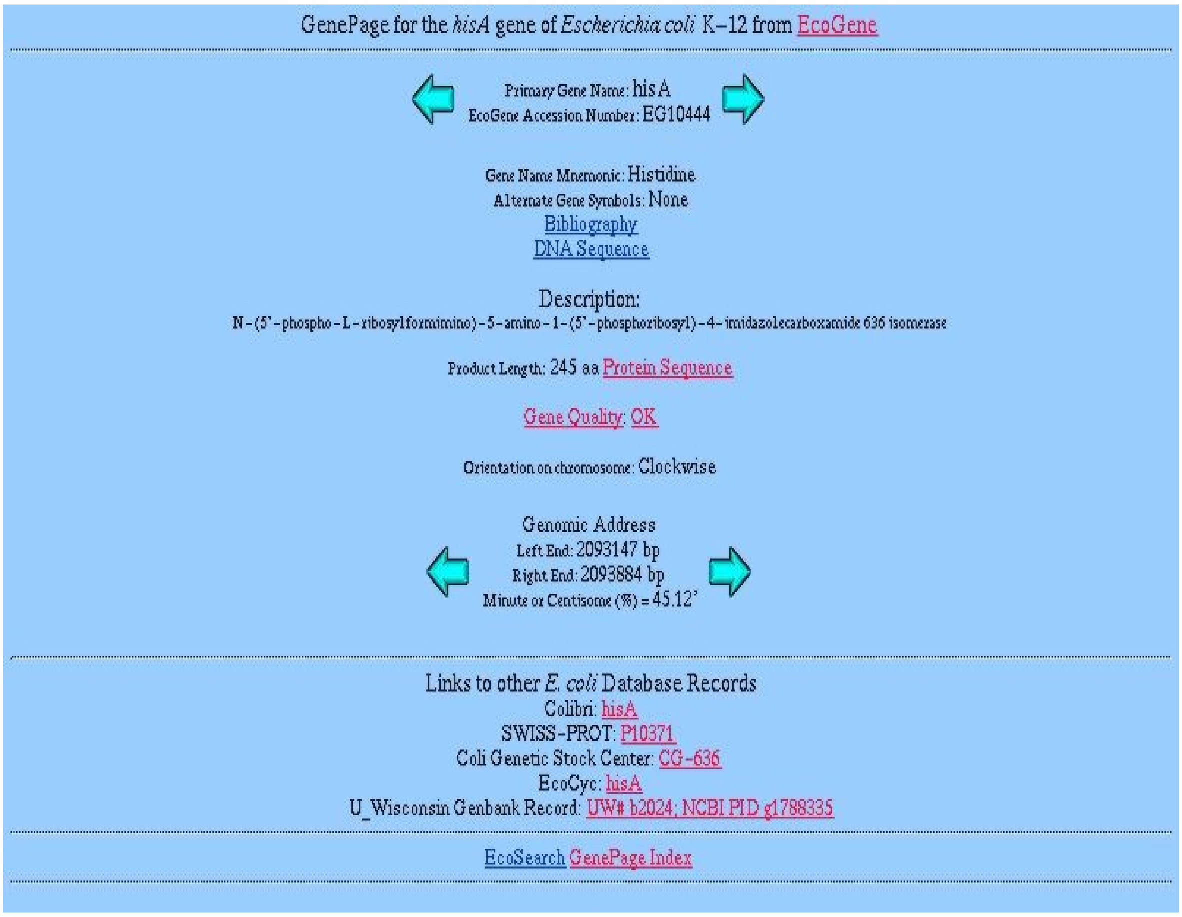


FIGURE A.10: GenePage for the a sample EcoGene entry.

A.27 EID: The Exon–Intron Database

Provider and access. The Exon–Intron Database (EID) is maintained at Harvard University at Cambridge/USA. It is accessible at: `http://mcb.harvard.edu/gilbert/EID/`.

Contents. EID contains information about protein-coding intron-containing genes. It provides text references to GenBank (cf. A.36).

Data modelling, data storage, and data acquisition. EID consists of three flat files. One file stores the DNA sequences in FASTA format. It is directly constructed out of the GenBank files. A second file stores the protein sequences in FASTA format. This file is generated by translating all sequences of the first file. Thus every entry storing information about a gene or protein starts with a ">" sign as required by the FASTA program. The third file contains GenBank header information for all sequences in the first two files. The header information of each entry is extracted from the header portion of the corresponding GenBank entry. See [EIDa] for more details.

Summarising, data are derived from GenBank by parsing special files from GenBank flat file release.

Data retrieval and query answering. There are no fixed-form and no ad-hoc query interfaces. The data can only be downloaded: Each of the three files are available in UNIX and MS Windows zipped format. An entire distribution including tools, markers, and the files is also available for download.

Size. EID contains information about more than 51 000 genes. The uncompressed files require currently about 450 MB. As of Winter 2000 the entire distribution including tools, markers, and the files requires 800 MB. See the EID homepage for up-to-data information.

Literature on EID:

[EIDa] S. Saxonov et al. EID: the Exon-Intron Database – an Exhaustive Database of Protein-Coding Intron-Containing Genes. *Nucleic Acids Research*, 28(1):185-190, 2000.

[EIDb] `http://mcb.harvard.edu/gilbert/EID/`

A.28 EMBL: The European Molecular Biology Laboratory Sequence Database

Provider and access. The European Molecular Biology Laboratory Sequence Database (EMBL) is maintained at the European Bioinformatics Institute (EBI) at Cambridge, UK. It can be accessed at: `http://www.ebi.ac.uk/embl/`

Contents. EMBL provides all publicly available nucleotide sequences in tight collaboration with GenBank (cf. A.36) and the DNA Data Bank of Japan (DDBJ, cf. A.22). Hypertext links to several external databases are provided.

Data modelling, data storage, and data acquisition. The EMBL database is internally implemented using the relational DBMS Oracle. There is no information about the underlying relational database schema available. In addition, the entire database is stored in many flat files consisting of explicitly structured entries. A sample entry of EMBL flat file release can be browsed at the EBI homepage.

Data are acquired primarily from the research community and the collaborating databases (see above). Several tools assist computer-based direct submission of data from laboratories.

Note: The three collaborating databases (DDBJ, EMBL, and GenBank) have defined a feature table that determines which information is added to the annotated sequences in all databases. In fact, the structure of the flat file release of each database is slightly different in syntax and semantics, but an entry in one of the collaborating databases contains exactly the same information as the corresponding entries in the other collaborating databases.

Data retrieval and query answering. EMBL can be accessed via a simple fixed-form query interface. There is no *ad-hoc* query interface. EMBL data can also be accessed using a local Sequence

Retrieval System (SRS, cf. Chapter 4.3.7) server at the EBI.

There is also a classification of the data by organism that can be browsed to retrieve single data items.

In addition, the entire database can be downloaded via FTP at: `ftp://ftp.ebi.ac.uk/pub/databases/embl/`

The flat file release consists of 17 divisions each consisting of one or more files (see [EMBLb] for details).

Size. Detailed information can be obtained at [EMBLc].

Literature on EMBL:

[EMBLa] W. Baker et al. The EMBL Nucleotide Sequence Database. *Nucleic Acids Research*, 28(1):19-23, 2000.

[EMBLb] `http://www.ebi.ac.uk/embl/Documentation/User_manual/user_manual.html`

[EMBLc] `http://www3.ebi.ac.uk/Services/DBStats/`

A.29 EMGLib: The Enhanced Microbial Genomes Library

Provider and access. EMGLib can be accessed through two databases installed on WWW servers:

- Server at the Bioinformatic Pole of Lyon (PBIL):
  `http://pbil.univ-lyon1.fr/emglib/emglib.html`
- Server at the INRA Microbial Genetic Laboratory:
  `http://locus.jouy.inra.fr/micado`

EMGLib is maintained by both research groups.

Contents. EMGLib contains all procaryotic genomes completely sequenced and the yeast genome. This data is annotated with some improvement compared to the other common sequence databases such as GenBank (cf. A.36) or EMBL (cf. A.28).

The database provide hypertext links to SWISS-PROT (cf. A.103) and GenBank (cf. A.36).

Data modelling, data storage, and data acquisition. EMGLib originally is a flat file database. The structure of the data entries are similar to the GenBank format (as GenBank is the most important source for data acquisition). Some corrections and additions to the original GenBank genome entries are performed. E.g. a new line type ”Codon Adaptation Index” (CAI) value is added under the keyword ”/CAI”, cross-references pointing to SWISS-PROT under the keyword ”/db_xref”, etc. (see [EMGLa] for further information).

The implementation of the two web servers is different:

- at PBIL EMGLib is implemented using a ”home-made” DBMS called ACNUC. This system is also a retrieval system for the nucleotide sequence and protein sequence databases. ACNUC consists of indexed flat files. See [EMGLc] for further information on ACNUC.
- at INRA EMGLib is implemented using a networked relational database called MICADO (formerly known as MadBase), which uses the commercial DBMS Oracle and for a portable version the ACEDB software. See [EMGLb] for more detailed information on MICADO (respectively MadBase).

Sequence data are acquired from the genome division of GenBank and the NRSub database.

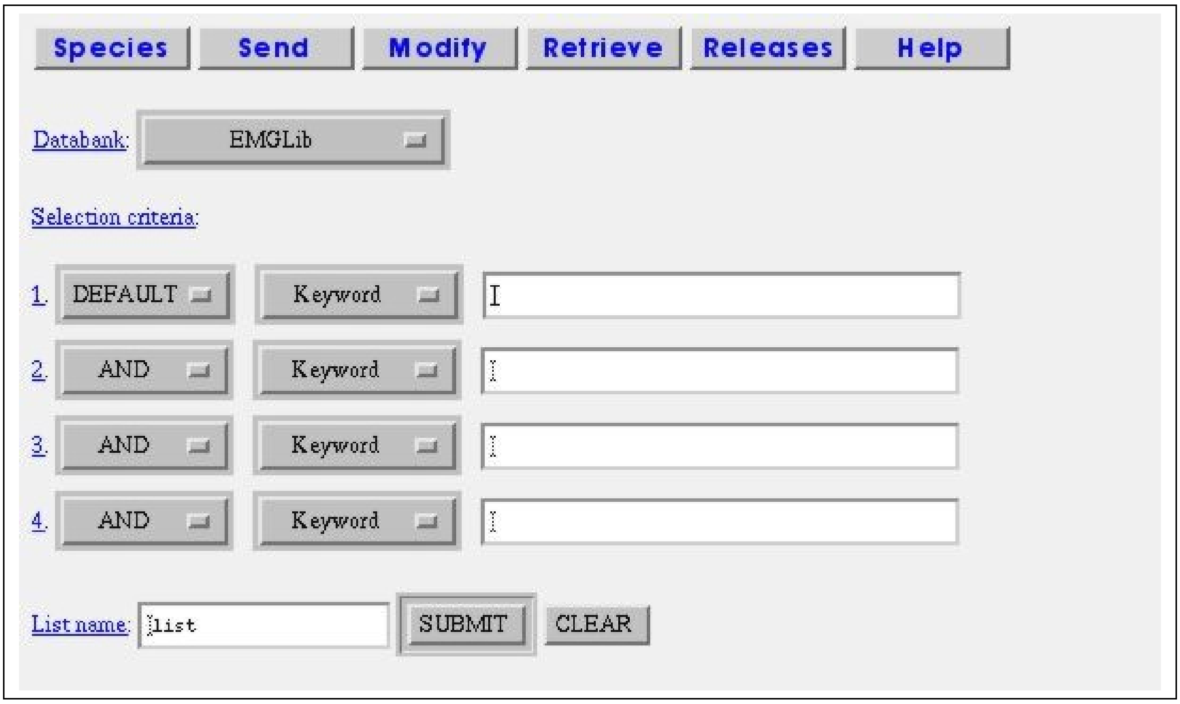


FIGURE A.11: Complex fixed-form query interface of EMGLib at PBIL.

Data retrieval and query answering. There are three simple interfaces provided by ACNUC to query keywords against a special line type (”keyword”, ”sequence name”, and ”accession number”). A more complex fixed-form query interface at PBIL integrates local copies of several flat file databases such as GenBank, EMBL, SWISS-PROT, etc. Search can be performed simultaneously against four line types of the databases (see Figure A.11). The results are represented in raw ASCII format with embedded hypertext links. EMGLib can also be accessed through the Sequence Retrieval System (SRS, cf. Chapter 4.3.7) at the PBIL home page.

At INRA EMGLib data can be accessed directly through a simple fixed-form query interface or indirectly following hypertext links to HTML representations of the browsed entries. Access to text only representations is also provided. The standard analysis tools such as BLAST and FASTA are within the database and accessible at the INRA home page.

Additional an EMGLib release is also available for download via FTP. This release contains:

- The flat version of EMGLib in GenBank format.
- The ACNUC index tables
- Two help files used respectively with the Query and Query_win programs

The architectures presently supported are: Sun/SPARC. IBM RS/6000, DEC Alpha, Silicon Graphics, HP/UX, Macintosh, and Windows 95

Size. No information available. Both servers are updated at each new GenBank, EMBL, SWISS-PROT, and PIR release.

Literature on :

[EMGLa] G. Perriere, P. Bessieres, and B. Labedan. EMGLib: the Enhanced Microbial Genomes Library. *Nucleic Acids Research*, 28(1):68-71, 2000.

[EMGLb] V. Biaudet, F. Samson, P. Bessieres. MadBase: a Networked rel. Database for Microbial Genomes. Available at: `http://locus.jouy.inra.fr/cgi-bin/genmic/madbase\protect\unhbox\voidb@x\kern.06em\vbox{\hrulewidth.3em}home.pl`

[EMGLc] `http://pbil.univ-lyon1.fr/databases/acnuc.html`

A.30 ENZYME

Provider and access. The ENZYME database is developed and maintained at the Swiss Institute of Bioinformatics, Geneva. It can be accessed at: `http://www.expasy.ch/enzyme/`

Contents. The ENZYME database contains information related to the nomenclature of enzymes. For each type of enzyme it lists several pieces of information such as EC number, recommended name, catalytic activity, etc. [ENZa]. Hypertext links to external databases such as among others SWISS-PROT (cf. A.103) and PROSITE (cf. A.85) are provided.

Data modelling, data storage, and data acquisition. ENZYME is a flat file database consisting of entries. An entry is explicitly structured using defined line types (indexes), each with its own format . For standardisation proposes the format of ENZYME follows as closely as possible that of the SWISS-PROT (cf. A.103) and EMBL (cf. A.28) sequence databases. A sample entry is shown in [ENZa].

In addition to the plain text file there is a flat file version structured using ANS.1 [5]. The ANS.1 specification of the ENZYME database is available at: `ftp://ftp.expasy.org/databases/enzyme/asn/enzspec.asn`

Data are acquired from the International Union of Biochemestry and Molecular Biology (IUBMB) and the scientific literature. In addition, a tight cooperation with SWISS-PROT (cf. A.103) ensures data exchange with another database.

Data retrieval and query answering. ENZYME can be queried via a simple fixed form query interface. There is no *ad-hoc* query interface. A second possibility to query keywords against searchable indexes is to access ENZYME via SRS (cf. Chapter 4.3.7).

The enzymes stored in ENZYME are structured in a hierarchy according to the EC numbers. The user can browse through this hierarchy to retrieve single data items.

In addition, the flat files (ENZYME-format and ANS.1 format) can be downloaded via FTP at: `ftp://ftp.expasy.org/databases/enzyme/`

Size. Currently the entire flat file in ENZYME-format requires approximately 1.5 MB where as the entire ANS.1 version of ENZYME requires about 4 MB.

Literature on ENZYME:

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

[ENZb] `http://www.expasy.ch/txt/enzuser.txt`

A.31 EPD: The Eukaryotic Promotor Database

Provider and access. The Eukaryotic Promoter Database EPD was designed and developed at the Weizmann Institute of Science in Rehovot (Israel) and is currently maintained at the Swiss Institute of Bioinformatics in Lausanne. It can be accessed at: `http://www.epd.isb-sib.ch/`

Contents. EPD is an annotated non-redundant collection of eukaryotic promoters, for which the transcription start site has been determined experimentally. It provides hypertext links to external databases such as among others EMBL (cf. A.28), MGD (cf. A.62), and SWISS-PROT (cf. A.103).

Data modelling, data storage, and data acquisition. Most likely EPD is implemented as a flat file database. The flat file release contains of entries that are explicitly structured through line types (search indexes). A detailed description of the different line types used in EPD is given in [EPDb].

There is no information about data acquisition of EPD.

Data retrieval and query answering. EPD can be accessed via two different fixed-form query interfaces. Several search indexes can be searched by keywords. A combined search of several keywords against different search indexes using boolean operators is also provided. There is no *ad-hoc* query interface nor a browsable hierarchy for data retrieval.

The entire database can be downloaded as flat files via FTP. In addition, EPD can be accessed using the local Sequence Retrieval System (SRS, cf. Chapter 4.3.7) server.

Size. The current flat file of EPD requires approximately 2 MB.

Literature on EPD:

[EPDa] R. Périer et al. The Eukaryotic Promotor Database (EPD). *Nucleic Acids Research*, 28(1):302-202, 2000.

[EPDb] `http://www.epd.isb-sib.ch/current/usrman.html`

A.32 ExInt: The Exon/Intron Database

Provider and access. The Exon/Intron Database (ExInt) is maintained at the National University of Singapore/Singapore in collaboration with researchers from the USA and Brazil. It is accessible at: `http://intron.bic.nus.edu.sg/exint/exint.html`

Contents. ExInt stores information on the exon and intron structure of eukaryotic genes. Features of ExInt include, among others, intron nucleotide sequences, amino acid sequences, and position of the introns at the amino acid level. ExInt provides hypertext links and textual references to GenBank (cf. A.36).

Data modelling, data storage, and data acquisition. The ExInt database is build as a text file in FASTA citepeli88 format. Each entry contains eight fields of information (see [EXINTa]). Additional to the entire ExInt file there are four more files containing only special data of ExInt:

- a file containing only data about predicted introns
- a file containing only data about experimentally defined introns
- a file containing only data about organelle introns
- a file containing only data about nuclear introns

Data are generated from GenBank (cf. A.36) using specialised parsing tools. See [EXINTa] for further information.

Data retrieval and query answering ExInt can be searched for similarity by a reference sequence (using BLAST [4]) or by keywords. Data can be retrieved by download of one of the five files (entire database, predicted introns, etc. see above). This requires registration but is free of charge.

There are no *ad-hoc* query or more complex fixed-form query interfaces. There is also no capability to browse the data.

Size. The current version of ExInt is derived from GenBank release 116 (February 2000). Detailed information about the size of the file is no provided.

Literature on ExInt:

[EXINTa] M. Sakharkar et al. ExInt: an Exon/Intron Database. *Nucleic Acids Research*, 28(1):191-192, 2000.

[EXINTb] `http://intron.bic.nus.edu.sg/exint/exint.html`

A.33 FIMM

Provider and access. The FIMM (Functional molecular IMMunology) database is maintained by the Kent Ridge Labs at Singapore and can be accessed at: `http://sdmc.krdl.org.sg:8080/fimm/`.

Contents. FIMM contains data relevant to functional molecular immunology. FIMM is cross-linked by hypertext links with several other databases, such as MedLine (cf. A.59), OMIM (cf. A.72), IMGT (cf. A.47), GenBank (cf. A.36), SWISS-PROT (cf. A.103), etc.

Table 1. FIMM data dimensions

Protein antigens	Peptides	MHC (HLA)	Diseases	Sources
ID	ID	ID	ID	PMID
Date	Date	Date	Date	Title
Name	Sequence	Name	Name	Authors
Aliases	MHC	Aliases	Aliases	Journal
Features	T-cell epitope	Sequence	Etiology	Links
Sequence	Naturally processed	Disease assoc.	Description	-PubMed
Links	Peptide binding	Binding pockets	Disease association	
-Peptides	Links	Binding sites CD8	Links	
-Diseases	-Prot. antigens	Binding sites TcR	-Protein antigens	
-Sources	-MHC	Alignments	-MHC	
-SWISSPROT	-Sources	Sources	-Sources	
-Genbank	-MHCPEP	Links	-OMIM	
	-SYFPAETHY	-Peptides	-GeneCards	
	- HIVdB	-Diseases	-Genatlas	
	- SWISSPROT	-IMGT database		

FIGURE A.12: The current FIMM data model has five dimensions (or views): protein antigens, peptides, MHC (HLA), diseases, and publication sources (taken from [FIMMb]).

Data modelling, data storage, and data acquisition. FIMM is a flat file repository that consist of data entries. The current "data model"has five views as shown in Figure A.12.

Data in FIMM are mainly acquired from the scientific literature and other databases.

Data retrieval and query answering. FIMM is searchable via a fixed-form query interface. The database can be queried for specific information within one of the particular views mentioned above (see also Figure A.12).

There is no *ad-hoc* query interface and no flat file release for download.

Several analysis tools such as BLAST [4] are with the database.

Data cannot be browsed or downloaded via FTP as flat files.

Size. Currently FIMM contains 5428 entries. Information on the size of the files are not provided.

Literature on FIMM:

[FIMMa] C. Schönbach et al. FIMM, a Database of Functional Molecular Immunology. *Nucleic Acids Research*, 28(1):222-224, 2000.

[FIMMb] `http://sdmc.krdl.org.sg:8080/fimm/`

A.34 FlyBase

Provider and access. FlyBase is maintained by the FlyBase Consortium, which is an international collaboration between several academic institutes in the USA and Europe. The data in FlyBase can be accessed at independent, but hyper-linked, WWW sites:

- FlyBase Consortium: `http://flybase.bio.indiana.edu/`
- Berkeley Drosophila Genome Project: `http://fruitfly.berkeley.edu/`
- European Drosophila Genome Project: `http://edgp.ebi.ac.uk/`

Contents. FlyBase stores genetic and genomic data about the *Drosophila* organisms. It provides hypertext links to several other databases such as Genbank (cf. A.36) and SWISS-PROT/TrEMBL (cf. A.103).

Data modelling, data storage, and data acquisition. The data are stored using the relational DBMS Sybase. A comprehensive and detailed description of the underlying schema is available at [FlyBb].

An underlying ontology is developed in collaboration with SGD (cf. A.100) and MGD (cf. A.62).

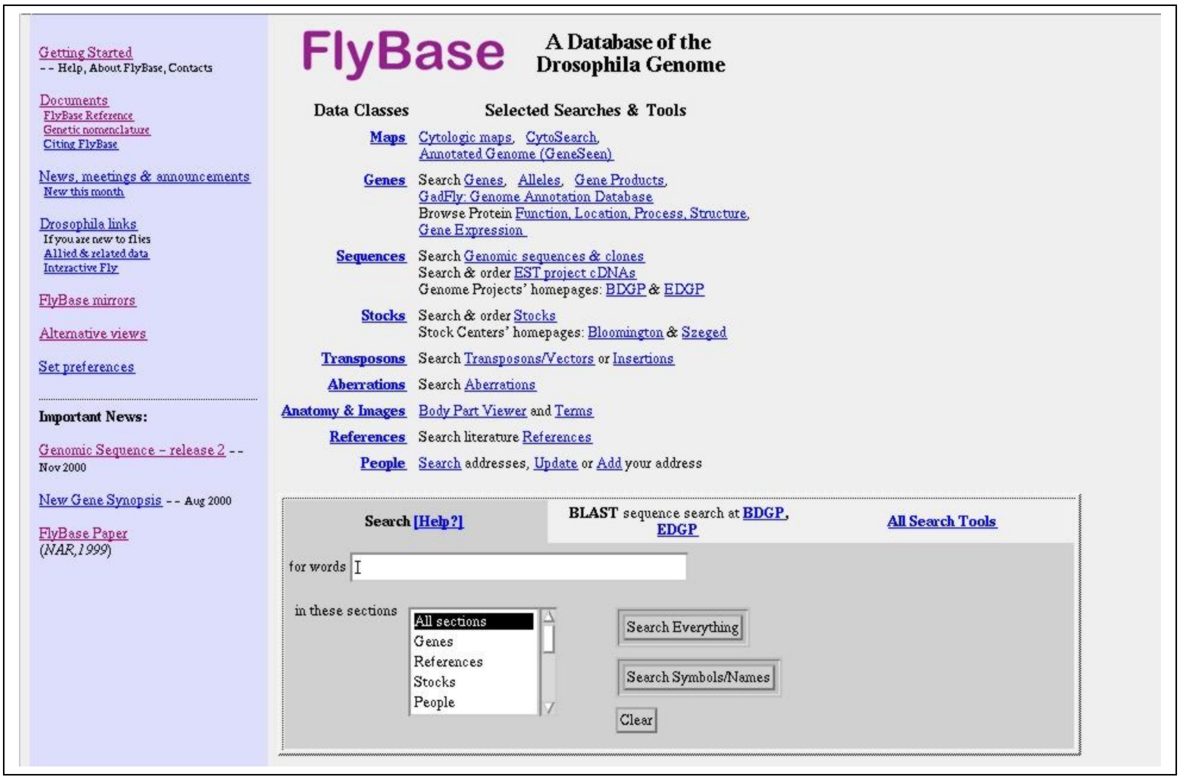


FIGURE A.13: Homepage of FlyBase.

Data are acquired from the scientific literature, the research community (submissions via a web-based fill-in form), and other external databases.

Data retrieval and query answering The most simple way to query FlyBase is to use the all-text interface. Keywords can be searched in the entire database or in determined datasets.

FlyBase data are organised into a variety of data classes for more complex queries and browsing. For these classes fixed-form query interfaces, that permit field-specific searches, combinatorial queries and menu-driven selection of controlled vocabularies, are available. Figure A.13 shows the starting page of FlyBase. The different classes and their query tools can be seen in the top centre. The simple all-text query form is located below. There is no *ad-hoc* query interface.

Data can additionally be retrieved through download by FTP.

Size. No information available.

Literature on FlyBase:

[FlyBa] The FlyBase Consortium. The FlyBase Database of the Drosophila Genome Projects and Community Literature. *Nucleic Acids Research*, 28(1):85-88, 2000.

[FlyBb] `http://flybase.bio.indiana.edu/.data/docs/refman/refman-sections.html`

A.35 GDB: The Human Genome Database

Provider and access. The Human Genome Database GDB is maintained at John Hopkins University in Baltimore, USA and is the official central repository for genomic mapping data resulting from the Human Genome Initiative. It is accessible at: `http://www.gdb.org`.

Size. The Human Genome Initiative is a worldwide research effort to analyse the structure of human DNA and determine the location and sequence of the estimated 100,000 human genes. In support of this project, GDB stores and curates data generated worldwide by those researchers engaged in the mapping effort of the Human Genome Project (HGP).

Hypertext links are provided to the IC sequence databases GenBank (cf. A.36), EMBL (cf. A.28), and DDBJ (cf. A.22), as well as OMIM (cf. A.72) and the Mouse Genome Database (MGD, cf. A.62).

Data modelling, data storage, and data acquisition. The GDB consists of three distinct databases (the Human Genome database HGD, the Citation Database, and the Human Genome Registry Database), which are all modelled and implemented using OPM on top of the relational DBMS Sybase.

All classes in the schemas of the three databases are subclasses of the class “Accessioned Object”. Attributes of this most general class, which are inherited each of its subclasses, are (among others): “Accession”, “ID”, “Submitter”, and “Name”.

The three main subclasses of “Accessioned” Object are “HGD” (genomic data), “Citation” (literature references), and “Registry” (people and organisations).

The OPM schema of GDB allows only four different attribute types: Text, Number, Restricted (i.e. the value must be one of a pre-defined list of values), and Reference (i.e. the attribute refers to another object in the database).

We will briefly investigate the schema of the HGD. Detailed information on the schema HGD and both other databases is accessible at [GDBb].

Four major classes of the HGD hierarchy are:

- ”Biological Object”:
  This class holds information about biological functions associated with instances of other classes such as ”Genes”.

- ”Experiment Object”:
  This class groups all objects which represent experimental data about maps and their components.

- ”Variation Object”:
  This class includes all objects involved in representing genomic variation, both common and uncommon. Polymorphisms are represented by several objects, each adding different levels of information and related to the others in a specific manner.

- ”Mapping Objects”:
  This class stores all objects required to represent and display genomic maps.

Data are acquired from the community by submission via dedicated forms. Only laboratories, which are members of the Human Genome Project, can add data. Submissions of other research groups are not accepted. Each entry is owned by the submitter. Only the owner or GDB staff is allowed to update an entry in GDB.

Data retrieval and query answering. GDB provides several ways to execute queries and retrieve data:

1. Fixed-form queries:
   - A simple interface enables the user to search for GDB-ID, keyword or DNA sequence ID against either ”HGD”, ”Registry”, ”Citation”, or all of them.
   - Advanced query interfaces provide with he possibility to use customised, sequence-based, or generic search forms. Each form generates a view on subsets of the GDB schema.
2. Free-form queries:
   Although the OPM tools offer an interface to formulate *ad-hoc* queries in the OPM query language, there is no free-form query interface accessible via the GDB homepage.

In addition, there are several criteria to browse the data in GDB.

Data are not offered in flat file format by GDB.

Size. As of October 2000 the GDB contains over 15 million objects, including 11 155 genes, 800 pseudo-genes, and 38797 putative genes.

Literature on GDB:

[GDBa] S. Letovsky et al. GDB: the Human Genome Database. *Nucleic Acids Research*, 28(1):289-291, 2000.

[GDBb] `http://www.gdb.org/gdb/schema.html`

A.36 GenBank

Provider and access. GenBank is maintained at the National Center for Biotechnology Information (NCBI) at Bethesda, USA. It can be accessed at: `http://www.ncbi.nlm.nih.gov/`

```
LOCUS       SCU49845     5028 bp    DNA             PLN       21-JUN-1999
DEFINITION  Saccharomyces cerevisiae TCP1-beta gene, partial cds, and Axl2p
            (AXL2) and Rev7p (REV7) genes, complete cds.
ACCESSION   U49845
VERSION     U49845.1  GI:1293613
KEYWORDS    .
SOURCE      baker's yeast.
  ORGANISM  Saccharomyces cerevisiae
            Eukaryota; Fungi; Ascomycota; Hemiascomycetes; Saccharomycetales;
            Saccharomycetaceae; Saccharomyces.
REFERENCE   1  (bases 1 to 5028)
  AUTHORS   Torpey,L.E., Gibbs,P.E., Nelson,J. and Lawrence,C.W.
  TITLE     Cloning and sequence of REV7, a gene whose function is required for
            DNA damage-induced mutagenesis in Saccharomyces cerevisiae
  JOURNAL   Yeast 10 (11), 1503-1509 (1994)
  MEDLINE   95176709
REFERENCE   2  (bases 1 to 5028)
  AUTHORS   Roemer,T., Madden,K., Chang,J. and Snyder,M.
  TITLE     Selection of axial growth sites in yeast requires Axl2p, a novel
            plasma membrane glycoprotein
  JOURNAL   Genes Dev. 10 (7), 777-793 (1996)
  MEDLINE   96194260
REFERENCE   3  (bases 1 to 5028)
  AUTHORS   Roemer,T.
  TITLE     Direct Submission
  JOURNAL   Submitted (22-FEB-1996) Terry Roemer, Biology, Yale University, New
            Haven, CT, USA
FEATURES             Location/Qualifiers
     source          1..5028
                     /organism="Saccharomyces cerevisiae"
                     /db_xref="taxon:4932"
                     /chromosome="IX"
                     /map="9"
     CDS             <1..206
                     /codon_start=3
                     /product="TCP1-beta"
                     /protein_id="AAA98665.1"
                     /db_xref="GI:1293614"
                     /translation="SSIYNGISTSGLDLNNGTIADMRQLGIVESYKLKRAVVSSASEA
                     AEVLLRVDNIIRARPRTANRQHM"
     gene            687..3158
                     /gene="AXL2"
     CDS             687..3158
                     /gene="AXL2"
                     /note="plasma membrane glycoprotein"
                     /codon_start=1
                     /function="required for axial budding pattern of S.
                     cerevisiae"
                     /product="Axl2p"
                     /protein_id="AAA98666.1"
                     /db_xref="GI:1293615"
                     /translation="MTQLQISLLLTATISLLHLVVATPYEAYPIGKQYPPVARVNESF
                     TFQISNDTYKSSVDKTAQITYNCFDLPSWLSFDSSSRTFSGEPSSDLLSDANTTLYFN
                     VILEGTDSADSTSLNNTYQFVVTNRPSISLSSDFNLLALLKNYGYTNGKNALKLDPNE
                     VFNVTFDRSMFTNEESIVSYYGRSQLYNAPLPNWLFFDSGELKFTGTAPVINSAIAPE
                     TSYSFVIIATDIEGFSAVEVEFELVIGAHQLTTSIQNSLIINVTDTGNVSYDLPLNYV
                     YLDDDPISSDKLGSINLLDAPDWVALDNATISGSVPDELLGKNSNPANFSVSIYDTYG
                     DVIYFNFEVVSTTDLFAISSLPNINATRGEWFSYYFLPSQFTDYVNTNVSLEFTNSSQ
                     DHDWVKFQSSNLTLAGEVPKNFDKLSLGLKANQGSQSQELYFNIIGMDSKITHSNHSA
                     NATSTRSSHHSTSTSSYTSSTYTAKISSTSAAATSSAPAALPAANKTSSHNKKAVAIA
                     CGVAIPLGVILVALICFLIFWRRRRENPDDENLPHAISGPDLNNPANKPNQENATPLN
                     NPFDDDASSYDDTSIARRLAALNTLKLDNHSATESDISSVDEKRDSLSGMNTYNDQFQ
                     SQSKEELLAKPPVQPPESPFFDPQNRSSSVYMDSEPAVNKSWRYTGNLSPVSDIVRDS
                     YGSQKTVDTEKLFDLEAPEKEKRTSRDVTMSSLDPWNSNISPSPVRKSVTPSPYNVTK
                     HRNRHLQNIQDSQSGKNGITPTTMSTSSSDDFVPVKDGENFCWVHSMEPDRRPSKKRL
                     VDFSNKSNVNVGQVKDIHGRIPEML"
     gene            complement(3300..4037)
                     /gene="REV7"
     CDS             complement(3300..4037)
                     /gene="REV7"
                     /codon_start=1
                     /product="Rev7p"
                     /protein_id="AAA98667.1"
                     /db_xref="GI:1293616"
                     /translation="MNRWVEKWLRVYLKCYINLILFYRNVYPPQSFDYTTYQSFNLPQ
                     FVPINRHPALIDYIEELILDVLSKLTHVYRFSICIINKKNDLCIEKYVLDFSELQHVD
                     KDDQIITETEVFDEFRSSLNSLIMHLEKLPKVNDDTITFEAVINAIELELGHKLDRNR
                     RVDSLEEKAEIERDSNWVKCQEDENLPDNNGFQPPKIKLTSLVGSDVGPLIIHQFSEK
                     LISGDDKILNGVYSQYEEGESIFGSLF"
BASE COUNT     1510 a   1074 c    835 g   1609 t
```

FIGURE A.14: Sample entry of GenBank (taken from [GenBc]).

Genbank/

Contents. GenBank provides all publicly available nucleotide sequences in tight collaboration with the EMBL Data Library (cf. A.28) and the DNA Data Bank of Japan (DDBJ, cf. A.22). It provides hypertext links to several other data bases (most of these links via Entrez (cf. Chapter 4.3.4)) such as SWISS-PROT (cf. A.103), PubMed (cf. A.90), OMIM (cf. A.72), etc.

Data modelling, data storage, and data acquisition. For internal data management, GenBank is implemented using the relational DBMS Sybase. The underlying relational data model is presented in [GenBb] (cf. [43]). For release, the data is converted into several different flat file formats: the "traditional" GenBank flat file format (see Figure A.14 for a sample entry), an ASN.1 structured flat file format, BLAST format, etc. All these files are structured explicitly using search indexes. The

ANS.1 formatted files are required for the Entrez system (for details cf. Chapter 4.3.4).

Data are acquired from the scientific community and the two collaborating nucleotide sequence databases EMBL (cf. A.28) and DDBJ (cf. A.22). Submissions from researchers is piped through two submission tools called *BankIt* and *Sequin*.

Note: The three collaborating databases (DDBJ, EMBL, and GenBank) have defined a feature table that determines which information is added to the annotated sequences in all databases. In fact, the structure of the flat file release of each database is slightly different in syntax and semantics, but an entry in one of the collaborating databases contains exactly the same information as the corresponding entries in the other collaborating databases.

Data retrieval and query answering. GenBank can be searched against a keyword (simple fixed-form query interface). The search results are provided and managed by Entrez (cf. Chapter 4.3.4) the NCBI's integrated database retrieval tool. There is no *ad-hoc* query interface, nor a possibility to browse the data to retrieve single data items.

Data are available as structured flat files in several different formats (see above).

Size. The current size of GenBank is provided at: `http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html`

Literature on GenBank:

[GenBa] D. Benson et al. GenBank. *Nucleic Acids Research*, 28(1):15-18, 2000.

[GenBb] H. Kitakami et al. Building and Search System for a Large-Scale DNA Database. In: S Schulze-Kremer, editor, *Advances in Molecular Bioinformatics*, pages 123-138, IOS Press, 1994.

[GenBc] `http://www.ncbi.nlm.nih.gov/Genbank/`

A.37 GIMS: The Genome Information Management System

Provider and access. The GIMS project is developing an object-oriented database of genome and associated functional information. The first public release is announced for Autumn 2000 at: `http://img.cs.man.ac.uk/gims` .

Contents. The main focus is on information about *S. cerevisiae*. Hypertext links to external databases are provided.

Data modelling, data storage, and data acquisition. In [GIMSa] Paton et al. present a collection of conceptual data models for genomic data. The data models are implementation-independent and

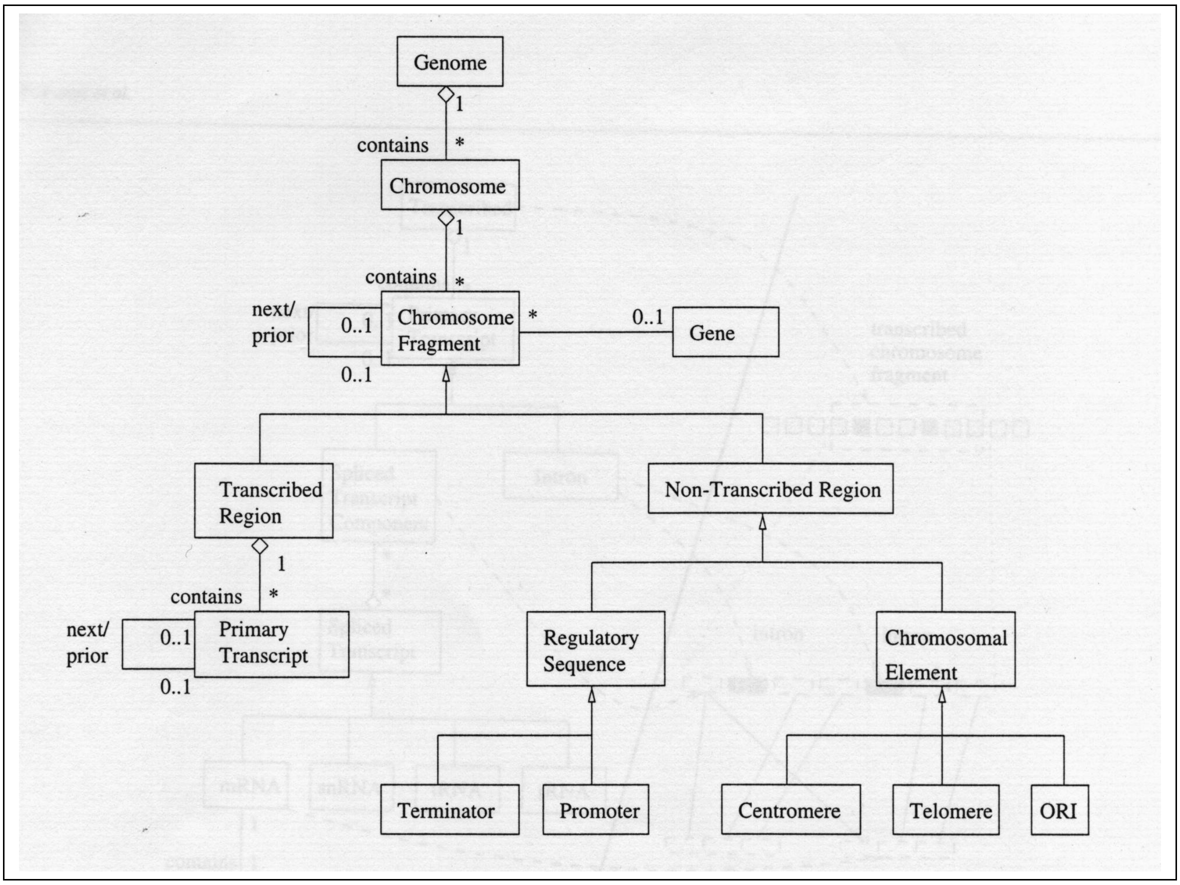


FIGURE A.15: Basic Schema digram for genomic data ([GIMSa])

developed using UML [8]. The models consist of a basic schema diagram for genomic data (which models the core data) and other schema diagrams e.g. for protein interaction data (see Figures A.15 and A.16).

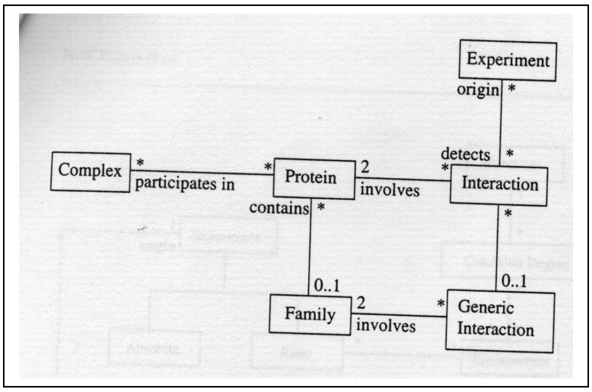


FIGURE A.16: Class diagram for protein interaction data ([GIMSa])

The schema diagram for genomic data is developed for modeling the fully sequenced genome of *S. cerevisiae*. Thus the current version presented in [GIMSa] models the alternate splicing in its simplest form – even well adequate for *S. cerevisiae*.

But it has to be extended if more complex organisms should be modelled. The schema is also not suitable for modeling mapping data (not yet fully sequenced genome data). Information on experiments or on literature are also not modelled so far. In [GIMSa] there is an abstract superclass suggested to allow literature or experimental details to be provided for all classes.

Other information models which are related to the basic model can be used to describe experiments and their results.

For example the class diagram for protein-protein interactions intersects with the basic diagram at the class "Protein". Detailed information about the interaction of two different proteins can be modelled as well as information about the experimental techniques used to detect the interaction. Also categories of interactions can be modelled.

The data models are mapped to classes in the ODMG compliant commercial object DBMS Poet.

The core data are principally acquired from MIPS (cf. A.63) and associated information is generally acquired via wrappers from remote sites.

Data retrieval and query answering. The implementation of a public interface is still in progress. The principal way of data access within GIMS is expected to be via fixed-form query interfaces.

Size. There is no information available. The reason for this might be that the first release is not yet published.

Literature on GIMS:

[GIMSa] N. Paton et al. Conceptual modeling of genomic information. *Bioinformatics*, 26(6):548-557, 2000.

[GIMSb] `http://img.cs.man.ac.uk/gims`

A.38 GSDB: The Genome Sequence DataBase

Provider and access. The database is maintained by the National Center for Genome Resources (NCGR) and freely available at: `http://www.ncgr.org`.

Contents. The Genome Sequence DataBase (GSDB) is a repository aiming at consisting all publicly available nucleotide sequences together with their associated biological and bibliographic information.

GSDB provides hypertext links to several other databases.

Data modelling, data storage, and data acquisition. The data are built upon a relational data model (see [GSDBb]) and stored using the commercial relational DBMS Sybase. Only Sybase data types are used except an "acc" datatype used for storing the "IC accession number" (implemented as char(10)). The values of some attributes are restricted to sets of explicit values, e.g. the value of the attribute "aa" of the Anticodon table (which stores the amino acid of the anticodon) is restricted to either one of the 20 amino acids that occur in proteins in the common three letter code or one of the values "Xxx", "TERM" or "OTHER". Every table has an attribute called "id" which corresponds to

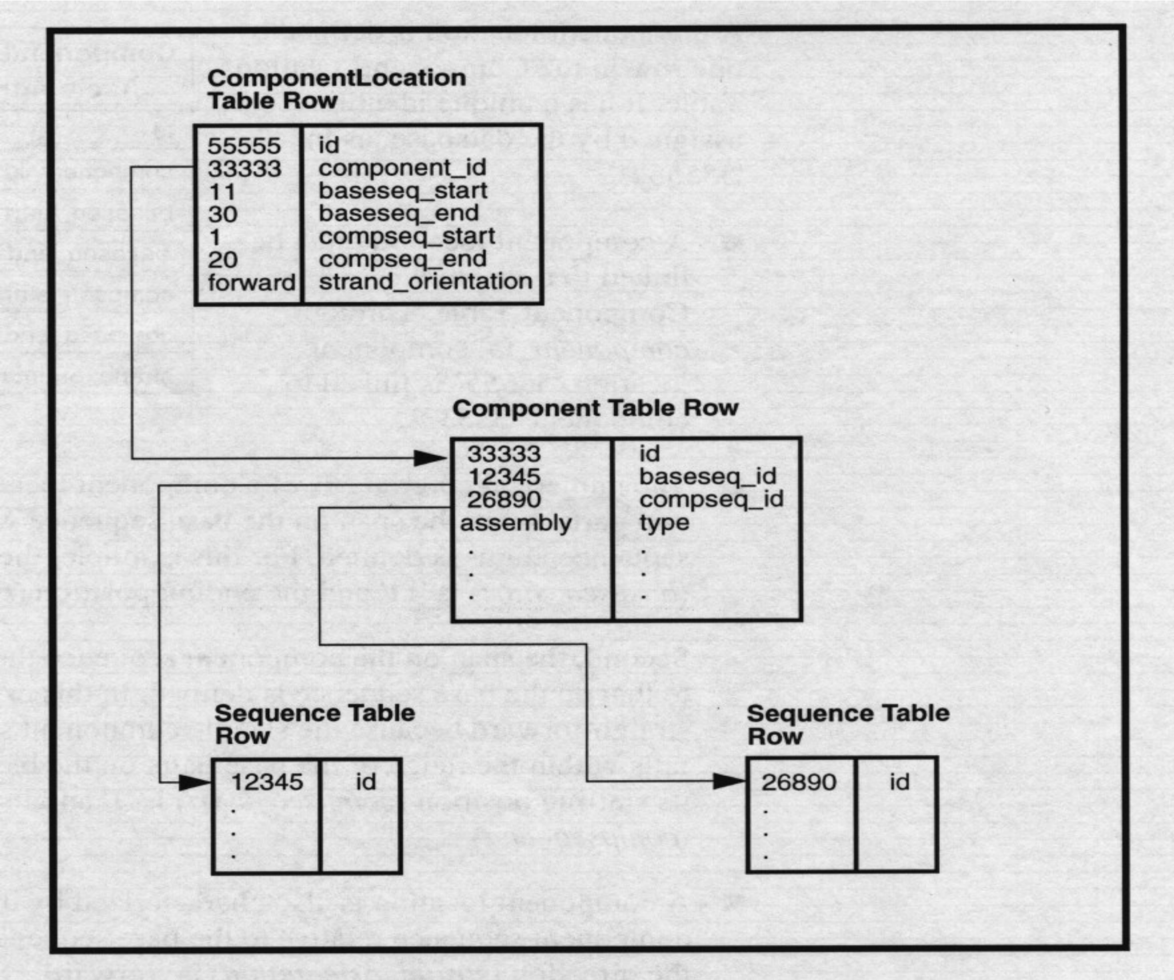


FIGURE A.17: Modelling of Sequence Alignments in GSDB: Example.

the SYBASE identity data type. Id attributes are generated automatically by the DBMS for each row of the table and are used as the primary key. Flexible and expressive mechanisms for representing biological data types are supported, for example:

- Multiple Sequence Alignments. (Multiple Sequence Alignments are from the modeling point of view correlations of one or more sequences with another sequence, here called "base sequence".) They are represented in GSDB by pairwise relationships between sequences. These relationships are represented by the table Component which links the base sequence with the other aligned sequences. The Component table is linked to the Component Location table that describes the position of the alignment by base pair position reference (multiple locations are possible) and to the Difference table that annotates the base pair anomalies between the two sequences in the alignment. Figure A.17 shows an example: a base sequence with the primary key "12345" is aligned to a component sequence "26890" . This relationship is represented through the Component table row with the primary key "33333". This row refers to row "55555" of the Component Location table, which models the location of the alignment according to both sequences.

- The data type of Sequence Confidence Data allows the owner of a sequence to represent a simple estimate of the accuracy of data submitted for that sequence. It is implemented through the Confidence table.

- Another data type is used for representing discontignous sequences for low-pass and sample sequencing strategies. Three sets are represented in GSDB: Segmented sequences (groups of sequences for which a place order between the segments is known but the distance between the segments may not known), STS map sets (groups of sequences for which the distance between the segments is usually known) and binned sequence sets (groups of sequences derived from a piece of DNA; order and distance information is not known).

- Links to other databases are supported mainly through the four tables Gene, Product, Taxon and Reference which are all linked to the DBLink table. Several other tables are involved in modeling links to other databases.

A homemade interface – the so called "Object Layer" – written in C++ handles all database access and communication for various applications. The DBMS handles all common database issues like security and ownership of the data. A flat file version of the data is produced for download. This flat file version conveys the logical structure of the underlying relational data model. See [GSDBb] for further information on the data model and its implementation.

Data are only acquired automatically (at night) from the IC databases GenBank (cf. A.36), EMBL (cf. A.28), and DDBJ (cf. A.22). Data submissions from researchers are not accepted. Data in GenBank, EMBL or DDBJ–format are converted by special parsers into the schema of GSDB. Manual review is provided for data exceptions which caused parse errors. Data stored in GSDB can be corrected or updated by the data owner (the user that inserted the data into GenBank) as well as by the GSDB staff.

Data retrieval and query answering. Data access is only possible either through the Object Layer (see above) or by using SQL. Thus fixed-form queries are as well supported as free-form/*ad-hoc* queries. The Object Layer also provides download of the flat file version via FTP. The following tools are available via the WWW-interface of NCGR to retrieve and process data from GSDB:

1. Fixed-form queries:

    - A tool called "Maestro" provides special views on the database. The user can search the database for terms such as organisms' names, gene names and keywords. Search can be performed on single words or combination of terms using boolean connections (the searching is limited to approximately 20 of the most commonly queried fields in the database).
    - "Excerpt" is a tool that allows researchers to identify and retrieve from GSDB interesting regions of sequences (e.g. protein coding region) instead of the entire sequences.
    - "Sequence Viewer" is a visualisation tool implemented in Java. This tool gives rise to navigate, examine GSDB sequences, and inspect their associated biological features.
    - Similarity searching of a query sequence against GSDB can be performed using BLAST [4].
    - "MAR-Finder" is a tool for detecting MARs in DNA sequences.

2. Free-form/*ad-hoc* queries:

   - A tool called "*Ad hoc*" can be used to query any field of the database. The tool allows the user to use SQL to directly formulate and evaluate queries against GSDB. The user has to register at first use.

Size. In October 1999 GSDB contained about 5 million sequences consisting each of about 4 billion nucleotides. The size of the database is doubling about every 9-12 months.

Literature on GSDB:

[GSDBa] C. Harger et al. The Genome Sequence DataBase. *Nucleic Acids Research*, 28(1):31-32.

[GSDBb] `http://www.ncgr.org/gsdb/schema.html`

A.39 GXD: The Gene Expression Database

Provider and access. The Gene Expression Database (GXD) is maintained at The Jackson Laboratory at Bar Harbor/USA and is accessible, together with MGD (cf. A.62) and MTB (cf. A.70) at: `http://www.informatics.jax.org/`

Contents. GXD is a resource of gene expression information for the laboratory mouse.

GXD is cross-linked (hypertext links) with several other databases, such as MGD (cf. A.62), MTB (cf. A.70), OMIM (cf. A.72), MEDLINE (cf. A.59), SWISS-PROT (cf. A.103), etc.

Data modelling, data storage, and data acquisition. GXD is implemented using the relational DBMS Sybase. A WWW interface provides access to the database using HTML-based forms combined with CGI scripts. For further information on the underlying schema see [GDXb].

GXD uses the same controlled vocabulary as MDG, developed by the Mouse Genome Informatics (MGI) Group at The Jackson Laboratory in collaboration with other database providers (MGD cf. A.62).

Data are acquired from the scientific literature and the research community.

Data retrieval and query answering. The data can be accessed via several fixed-form query interfaces.

GXD provides an interface for *ad-hoc* queries expressed in SQL. Users wishing to perform SQL queries against GXD must register first.

The data can also be downloaded together with the data of MGD in several flat files.

Several data items are accessible through browsing.

Size. No information available.

Literature on GXD:

[GXDa] M. Ringwald et al. GXD: A Gene Expression database for the Laboratory Mouse: Current Status and Recent Enhancements. *Nucleic Acids Research*, 28(1):115-119, 2000.

[GXDb] M. Ringwald et al. GXD: a Gene Expression database for the Laboratory Mouse. *Nucleic Acids Research*, 27(1):106-112, 1999.

[GXDc] `http://www.informatics.jax.org/mgihome/GXD/aboutGXD.shtml`

A.40 HDB: The Histone Database

Provider and access. The Histone Database is maintained by the National Center for Biotechnology, National Institute of Health, at Bethesda/USA. It is accessible at: `http://genome.nhgri.nih.gov/histones/`.

```
>HSPM3|HSPM3 histone H3 - garden pea. [ Pisum sativum ]|peaH3
ARTKQTARKSTGGKAPRKQLATKAARKSAPATGGVKKPHRFRPGTVALREIRKYQKSTEL
LIRKLPFQRLVREIAQDFKTDLRFQSSAVSALQEAAEAYLVGLFEDTNLCAIHAKRVTIM
PKDIQLARRIRGERA

>HSPM4|HSPM4 histone H4 - garden pea. [ Pisum sativum ]|peaH4
SGRGKGGKGLGKGGAKRHRKVLRDNIQGITKPAIRRLARRGGVKRISGLIYEETRGVLKI
FLENVIRDAVTYTEHARRKTVTAMDVVYALKRQGRTLYGFGG

>HSPG4|HSPG4 histone H4 - pig. [ Sus scrofa domestica ]|porH4
SGRGKGGKGLGKGGAKRHRKVLRDNIQGITKPAIRRLARRGGVKRISGLIYEETRGVLKV
FLENVIRDAVTYTEHAKRKTVTAMDVVYALKRQGRTLYGFGG
```

FIGURE A.18: Three sample entries of HDB.

Contents. The HDB is an annotated collection of all full-length sequences and structures of histone and non histone proteins containing the histone fold motif. Hypertext links to OMIM (cf. A.72), GenBank (cf. A.36), PDB (cf. A.75), and MMDB (cf. A.67) are provided.

Data modelling, data storage, and data acquisition. There is no detailed information about the data model and its implementation. Nevertheless, we suggest that HDB is a flat file repository. The reason for this is, that the results of the queries are presented like flat file entries (e.g. as shown in Figure A.18).

Data are acquired from Genbank (cf. A.36).

Data retrieval and query answering. A simple fixed-form query interface can be used for data retrieval. The results are displayed as shown in Figure A.18.

In addition, the data can be browsed, e.g. by histone family. The data is presented as a flat file with hypertext links (see above). These flat files can be downloaded.

There is no *ad-hoc* query interface.

Size. Currently HDB stores 1540 entries. There is no further information about the size of the database available.

Literature on HDB:

[HDBa] S. Sullivan et al. The Histone Database: a Comprehensive WWW Resource for Histones and Histone Fold-Containing Proteins. *Nucleic Acids Research*, 28(1):320-322, 2000.

[HDBb] `http://genome.nhgri.nih.gov/histones/`

A.41 HGBASE: The Human Genetic Bi-Allelic SEquences

Provider and access. The Human Genetic Bi-Allelic SEquences (HGBASE) is maintained and provided by an international collaboration of several academic and industrial institutes. It is freely accessible at: `http://hgbase.cgr.ki.se`

Contents. HGBASE gathers human all information about human gene-linked polymorphisms and other gene variations. Hypertext links and text references to the IC databases Genbank (cf. A.36), EMBL (cf. A.28), and DDBJ (cf. A.22) as well as to SWISS-PROT (cf. A.103) is usually provided for each data item.

Data modelling, data storage, and data acquisition. HGBASE is organised as a two-level system. Local storage and handling of individual records is performed using the Microsoft Access relational DBMS. Home made scripts transfer data to either a different relational database platform (Oracle) or a simple flat file format.

There is no information about the schema of the database.

The data are derived from various sources, including the scientific literature, submissions from the research community and other databases. A comprehensive list of all HGBASE data sources is provided by [HGBa].

Data retrieval and query answering. HGBASE can be searched by keyword (against all data in HGBASE without the sequence data). The database is also accessible through the Sequence Retrieval System (SRS, cf. Chapter 4.3.7). SRS provides several fixed-form and *ad-hoc* query interfaces. A third way to query the database is to search for sequence similarity using a reference sequence. The similarity search is carried out by the BLAST [4] program. In addition, final data presentation is always given as simple flat files for download.

There are no capabilities to browse the data.

Size. No information available.

Literature on HGBASE:

[HGBa] A. Brookes et al. HGBASE: a aatabase of SNPs and other variations in and around human genes. *Nucleic Acids Research*, 28(1):356-360, 2000.

[HGBb] `http://hgbase.cgr.ki.se`

A.42 HGMD: Human Genome Mutation Database

Provider and access. The Human Genome Mutation Database (HGMD) is maintained at the University of Wales Collage of Medicine in Cardiff, UK. It can be accessed at: `http://archive.uwcm.ac.uk/uwcm/mg/hgmd0.html`

Contents. HGMD represents a comprehensive collection of data on mutations underlying human inherited diseases.

Hypertext links to external databases such as GDB (cf. A.35) and OMIM (cf. A.72) are provided.

Data modelling, data storage, and data acquisition. There is no information available about the data management system of HGMD and the underlying data model. Most likely, HGMD is a flat file database.

Data are acquired from the scientific literature.

Data retrieval and query answering. HGMD is accessible via a simple fixed-form query interface (keyword search against the entire database). There is no *ad-hoc* query interface. The data cannot be browsed nor be downloaded via FTP.

Size. HGMD consists of 21 591 entries. There is no further information available.

Literature on HGMD:

[HGMDa] D. Cooper, E. Ball, and M. Krawczak. The Human Gene Mutation Database. *Nucleic Acids Research*, 26(1):285-287, 1998.

[HGMDb] `http://www.infobiogen.fr/services/dbcat`

A.43 The Homeodomain Resource

Provider and access. The Homeodomain Resource is maintained at the National Human Genome Research Institute NHGRI in Bethesda, USA. It can be accessed at: `http://genome.nhgri.nih.gov/homeodomain/`

Contents. The database is a collection of annotated protein sequence data and genomic information for the homeodomain protein family.

Hypertext links to external resources such as among others SWISS-PROT (cf. A.103), OMIM (cf. A.72), MMDB (cf. A.67), and PubMed (cf. A.90).

Data modelling, data storage, and data acquisition. The database was originally stored as a series of flat files and HTML files. These data are now re-implemented using the relational DBMS Sybase. The underlying relational schema is not documented.

Data are acquired from the scientific literature and other databases (e.g. OMIM, cf. A.72).

Data retrieval and query answering. There are several fixed-form query interfaces to retrieval data from the Homeodomain Resource but no *ad-hoc* queries can be performed.

The data of the Homeodomain Resource are not classified and thus there is no possibility to retrieve data items by browsing. Sequence data is available as flat files in FASTA [4] format for download via FTP. There is no possibility to retrieve the entire database as flat file(s).

Size. No information available.

Literature on the Homeodomain Resource:

[HRa] S. Banerjee, J. Ryan, and A. Baxevanis. The Homeodomain Resource: a prototype database for a large protein family. *Nucleic Acids Research*, 28(1):329-220, 2000.

[HRb] `http://genome.nhgri.nih.gov/homeodomain/`

A.44 HOX Pro

Provider and access. HOX Pro is maintained at Russian Academy of Sciences in St. Petersburg, Russia in collaboration with the Mt. Sinai Medical School at New York, USA. It can be accessed at: `http://www.mssm.edu/molbio/hoxpro/new/hox-pro00.html`

Contents. HOX Pro contains information about special genes called homeobox genes. It also provides information about genetic regulatory networks of homeobox genes.

Hypertext links to several other databases are provided, e.g. to PubMed (cf. A.90), EMBL (cf. A.28) and Transfac (cf. A.109).

Data modelling, data storage, and data acquisition. Although [HOXa] provides detailed information about the modelling of single data items such as networks, genes, etc. it is not really clear how HOX Pro is implemented. We suppose that it is a relational database.

There is no information available about data acquisition in HOX Pro.

Data retrieval and query answering. Data in HOX Pro can only be accessed via browsing.

There are no query interfaces at all, nor any flat files released for downloading.

Size. No information available.

Literature on HOX Pro:

[HOXa] A. Spirov, T. Bowler, and J. Reinitz. HOX Pro: a specialized database for clusters and networks of homeobox genes. *Nucleic Acids Research*, 28(1):337-340, 2000.

[HOXb] `http://www.mssm.edu/molbio/hoxpro/new/hox-pro00.html`

A.45 IDB/IEDB: The Intron Sequence and Intron Evolution Databases

Provider and access. The Intron Sequence Database (IDB) and the Intron Evolution Database (IEDB) are developed and maintained at the Indiana University at Bloomington, USA. Both databases can be accessed at: `http://nutmeg.bio.indiana.edu/intron/index.html`

Contents. IDB contains information about introns, exons, protein coding regions, and descriptive information about the according genes and source organisms. IEDB provides a statistical analysis of the exon and intron sequences catalogued in IDB as well as data concerning intron evolution [IDBa].

Both databases are cross-linked to SWISS-PROT (cf. A.103) and GenBank (cf. A.36) via hypertext links.

Data modelling, data storage, and data acquisition. The IDB and IEDB are stored in relational formats. A home-made DBMS manages the data storage. There is no information available about the database schemas. In addition, the databases are available as flat files each structured through search indexes.

Data are acquired from SWISS-PROT (cf. A.103) and GenBank (cf. A.36).

Data retrieval and query answering. The IDB and IEDB databases are only available for download via FTP at: `http://nutmeg.bio.indiana.edu/intron/index.html`

The databases are available in either plain text flat file format or in a relational format. The home-made DBMS application can also be downloaded. A web-based query interface is announced to be under construction.

Size. The plain text formatted flat files require 273.1 MB (IDB) and 32.5 MB (IEDB). The relational formatted files require 81.9 MB (IDB) and 6.8 MB (IEDB).

Literature on IDB and IEDB:

[IDBa] N. Schisler and J. Palmer. The IDB and IEDB: Intron Sequence and Evolution Database. *Nucleic Acids Research*, 28(1):181-184, 2000.

[IDBb] `http://nutmeg.bio.indiana.edu/intron/index.html`

A.46 The IMB Jena Image Library of Biological Macromolecules

Provider and access. The IMB Jena Image Library of Biological Macromolecules (IMB database) is maintained at the Institut für Molekulare Biotechnologie (IMB) at Jena, Germany. It can be accessed at: `http://www.imb-jena.de/IMAGE.html`

Contents. The IMB database contains data on the 3-dimensional structure of biomolecules (primarily proteins and nucleic acids). It integrates the PDB (cf. A.75) and the NDB (cf. A.71).

Hypertext links to several other databases are provided.

Data modelling, data storage, and data acquisition. There is no information available about the data management and the underlying data model of the IMB database. We assume that the IMB database is implemented as flat files. One file seems to provide access to all structure entries in either PDB or NDB by storing cross-links to the according entries. Annotated information seem to be managed in a separate file.

Data are acquired primarily from the external databases PDB (cf. A.75) and NDB (cf. A.71).

Data retrieval and query answering. The IMB database can be searched through two different fixed-form query interfaces. There is no *ad-hoc* query interface. Data are classified into several hierarchies. All these classifications can be browsed to retrieve single data items.

There is no flat file release of the IMB database and thus the data cannot be downloaded via FTP.

Several analysis and especially visualisation tools (e.g. RasMol [60]) are available at the IMB database homepage.

Size. No information available.

Literature on the IMB database:

[IMBa] J. Reichert. The IMB Jena Image Library of Biological Macromolecules. *Nucleic Acids Research*, 28(1):246-249, 2000.

[IMBb] `http://www.imb-jena.de/IMAGE.html`

A.47 IMGT: The International ImMunoGeneTics database

Provider and access. The International ImMunoGeneTics database (IMGT) consists of three databases. All databases was created by Marie-Paule Lefranc, Universite Montpellier II, CNRS, Montpellier/France and are accessible at: `http://imgt.cines.fr:8104`
or at: `http://imgt.cines.fr:8104/textes/IMGTindex.html`.

Contents. IMGT is an integrated database with expertly annotated sequences specialised in immunoglobulins, T-cell receptors and major histocompatiblity complexes (MHC). As mentioned above, it consists of three databases: LIGM-DB for immunoglobulins and T-cell receptors, MHC/HLA-DB and PRIMER-DB for immunoglobulins, T-cell receptors and and MHC-related primer — the later one is still in development.

The IMGT databases provide hypertext links to other databases, especially to general sequence databases such as EMBL (cf. A.28) or SWISS-PROT(cf. A.103).

Data modelling, data storage, and data acquisition. IMGT databases are modelled using an entity relationship model. Exceptions which are produced by the researchers who submit data are recorded in exception tables. These tables should avoid the introduction of uncontrolled inconsistent data.

An ontology was developed for formal specification of the terms to be used in the domain of immunogenetics and bioinformatics, called IMGT-ONTOLOGY. This allows the management of immunogenetic knowledge for all vertebrate species.

IMGT databases are managed by the relational DBMS Sybase.

Data are acquired from sequence databases (primariliy EMBL, cf. A.28), annotated and reviewed manually and then inserted in the databases.

Data retrieval and query answering. The contents of the IMGT databases can be obtained as flat files either by CD-ROM, e-mail subscription or internet FTP anonymous access.

IMGT data can also be accessed via the integrated environments SRS (cf. Chapter 4.3.7) and Entrez (cf. Chapter 4.3.4).

A fixed-form query interface available at the IMGT homepage provides a limited view on the data. There is no *ad-hoc* query interface available.

The data can be browsed via the IMGT homepage at: `http://imgt.cines.fr:8104`.

Size. In October 2000 LIGM-DB consists of 41 248 entries and HLA-DB of 1257 entries.

Literature on IMGT:

[IMGTa] M. Ruiz et al. IMGT, the International ImMunoGeneTics Database. *Nucleic Acids Research*, 28(1):219-221, 2000.

[IMGTb] `http://imgt.cines.fr:8104`

[IMGTc] V. Giudicelli et al. IMGT/LIGM-DB: A Systematised Approach for ImMunoGeneTics Database Coherence and Data Distribution Improvement. In *Proceedings of the 6th International Conference on Intelligent Systems in Molecular Biology (ISMB)*, 1998.

A.48 InBase: The Intein Database

Provider and access. InBase, the intein database is maintained at New England BioLabs, Inc. It can be accessed at: `http://www.neb.com/neb/inteins.html`

Contents. InBase contains a full intein registry. Inteins are protein splicing elements that mediate a self-catalytic protein splicing reaction. Each entry in InBase represents an intein and contains general information as well as detailed data annotation. Hypertext links to PubMed are provided.

Data modelling, data storage and data acquisition. There is no information available about the underlying data model of InBase and the data management system in use. We suppose that InBase is probably a collection of flat files.

Data are primarily acquired from the research community. A submission tool facilitates the acquisition.

Data retrieval and query answering. InBase provides no query interface at all. The database can only be browsed to retrieve single data items. There is no possibility to download the data. An amino acid sequence can be searched against InBase using BLAST [4].

Size. No information available.

Literature on InBase:

[InBa] F. Perler. InBase, the Intein Database. *Nucleic Acids Research*, 28(1):344-345, 2000.

[InBb] `http://www.neb.com/neb/inteins.html`

A.49 INTERACT

Provider and access. INTERACT is maintained by the University of Manchester at Manchester, UK and is accessible at: `http://bioinf.man.ac.uk`

Contents. INTERACT is a protein interaction database that stores both experimental data on interactions and derived results from bioinformatics analysis. Available interaction data mainly concern proteins in yeast.

Data modelling, data storage, and data acquisition. A user requirement analysis was carried out. This identified the most important questions with which researchers involved in protein interaction experimentation would query the database (see [INTAa] for details). This information was used to identify a set of use-cases like it is common in the field of object oriented software engineering. These use-cases played an important role in the design of the database schema because they indicated some of the information that would be needed in addition to the basic interaction data.

To model the database schema the Unified Modeling Language (UML) [8] was used. The data model is based on the assumption that an *Interaction* consists of two interacting *Proteins*. An *Interaction* may be described by many *Experiments*, which indicates the method and conditions the interaction was discovered under ([INTa]). Figure A.19 presents the class diagram using UML. [INTa] provides more information on the model and on the procedure of design.

The database has been implemented using the commercial object DBMS Poet, which follows the ODMG standard. The UML schema is mapped straightforward to a collection of classes in the ODMG Java binding.

Data are acquired from three locations ([INTAa]):

- The interactions tables provided by the MIPS database
- Scientist submission from a WWW form
- Submission from published literature

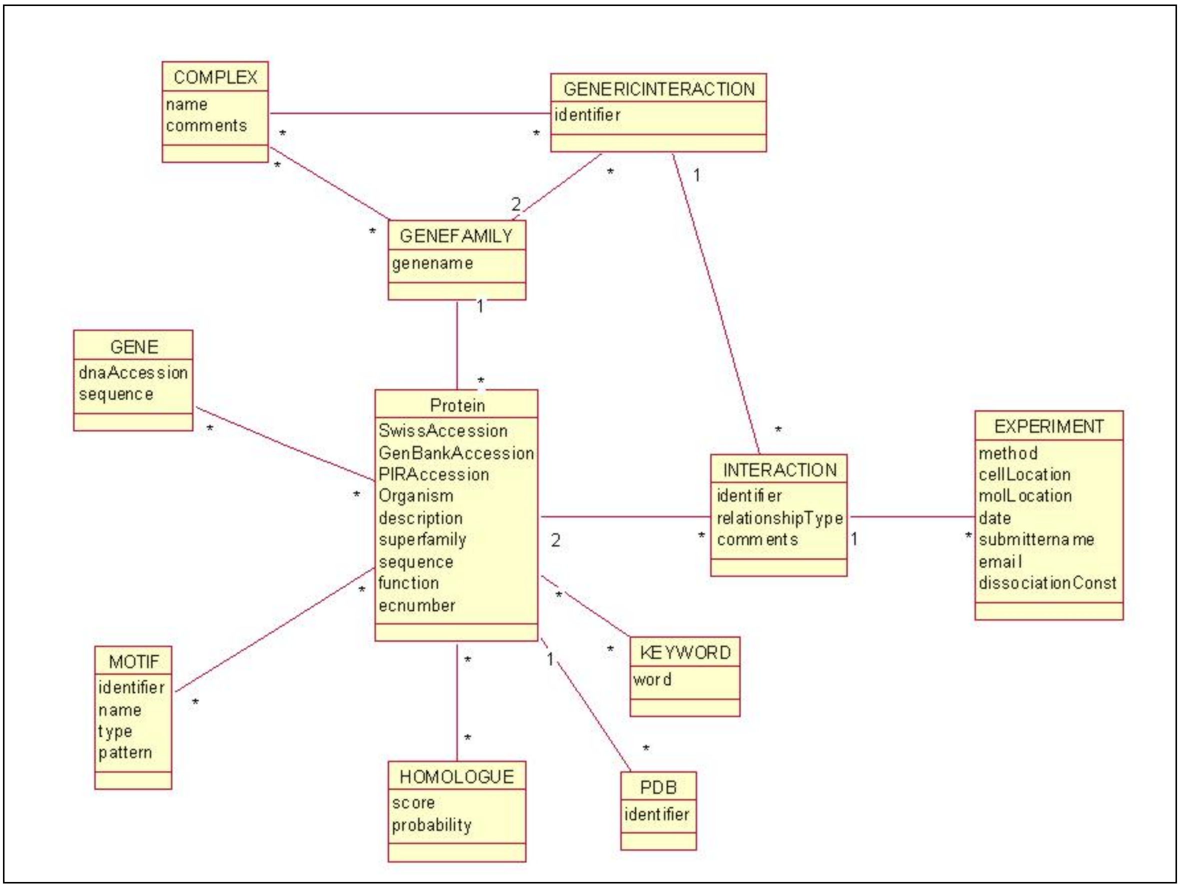


FIGURE A.19: UML class diagram of the INTERACT database [INTb]

Additionally the SWISS-PROT database is searched for much of the subsidiary information relating to a protein including e.g. links to MEDLINE (for reference information) and PROSITE (for motif information) databases. All these ways of data acquisition are carried out automatically by dedicated tools.

Data retrieval and query answering. INTERACT provides both *ad-hoc* and fixed-form queries ([INTAa]):

1. Fixed-form queries:
   The use-cases have been used to direct the development of a form based interface in Java. The top level of this interface consists of a list of questions. Selecting a question initiates an additional request for parameters and leads to a special view on the database. The results of the queries are presented in a form based browser.

2. *Ad-hoc* queries:
   OQL can be used to express declarative requests for information from the database.

There are also additional some analysis tools with the database. The user can access a clustering program to analyse the interactions and to get networks of interacting proteins. Methods based on graph theory algorithms can be used to detect any pathways of interaction between any two proteins. Another tool compares all of the proteins involved in yeast interactions to other genomes, using BLAST for homologue search.

Data cannot be downloaded as flat file(s). There is no capability to browse the INTERACT database.

Size. No information available.

Literature on INTERACT:

[INTAa] K. Eilbeck et al. INTERACT: an Object Oriented Protein-Protein Interaction Database. In *Proccedings of the 7th International Conference on Intelligent Systems in Molecular Biology (ISMB)*, 1999.

[INTAb] `http://bioinf.man.ac.uk`

A.50 InterPro

Provider and access. InterPro is maintained at the European Bioinformatics Institute (EBI) near Cambridge, UK. The InterPro project's web page is available at: `http://www.ebi.ac.uk/interpro/`

Contents. InterPro is an integrated database for protein families, domains, and functional sites, which try to merge the contents of several databases focusing on protein domain classification such as PROSITE (cf. A.85) and Pfam (cf. A.78). Each InterPro entry contains annotations and hypertext links back to the relevant database(s) where it is derived. In addition, each entry provides hypertext links to PubMed (cf. A.90).

Data modelling, data storage, and data acquisition. The data of InterPro are derived from various sources as flat files. These flat files are merged and dismantled. Internally all these data is managed using a relational DBMS. The underlying schema is available as figure in PDF-format at [INTPb].

The InterPro database is also released in two flat files in XML format. One file contains the core InterPro entries, the other file contains the protein matches with SWISS-PROT (cf. A.103). A corresponding Document Type Definition (DTD) file is also available.

Figure A.20 represents the InterPro data flow.

Data retrieval and query answering. InterPro can be accessed via a simple keyword search form (fixed-form query interface). Several search terms are allowed to be combined using boolean operators (AND, OR, NOT). There is no *ad-hoc* query interface nor a possibility to browse the database.

The XML structured files and the according DTD file can be downloaded via FTP at `ftp://ftp.ebi.ac.uk/pub/databases/interpro/`

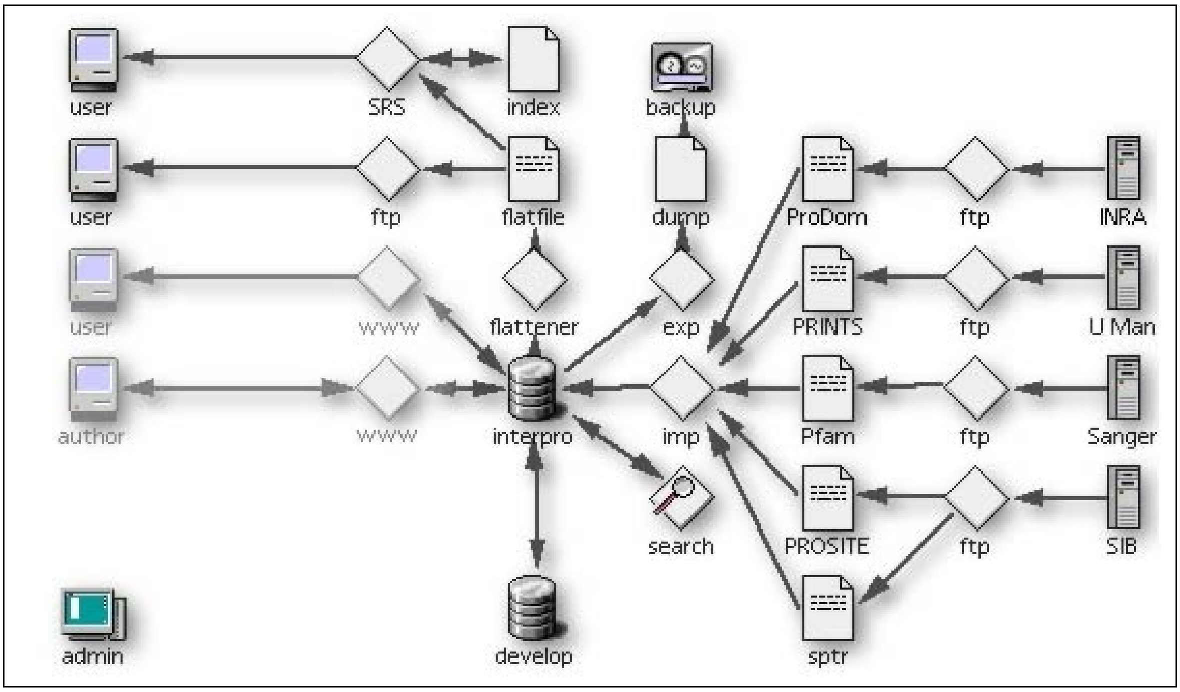


FIGURE A.20: InterPro data flow: data are acquired from flat files of external databases via FTP (right). After merging, these data are managed within a relational DBMS which is converted into an XML file or can be accessed directly via www (left).

In addition, InterPro can be accessed via SRS (e.g. via the local SRS server at the EBI).

Size. The zipped XML file with the core data of InterPro currently requires 2.3 MB.

Literature on InterPro:

[INTPa] R. Apweiler et al. The InterPro Database, an Integrated Documentation Resource for Protein Families, Domains, and Functional Sites. *Nucleic Acids Research*, 29(1):37-40, 2001.

[INTPb] `http://www.ebi.ac.uk/interpro/documentation.html`

[INTPc] `http://www.ebi.ac.uk/interpro/dataflow_scheme.html`

A.51 IXDB: Integrated X Chromosome Database

Provider and access. The Integrated X Chromosome Database (IXDB) is maintained at The Max-Planck-Institut for Molecular Genetic at Berlin, Germany and can be accessed at: `http://ixdb.mpimg-berlin-dahlem.mpg.de/`

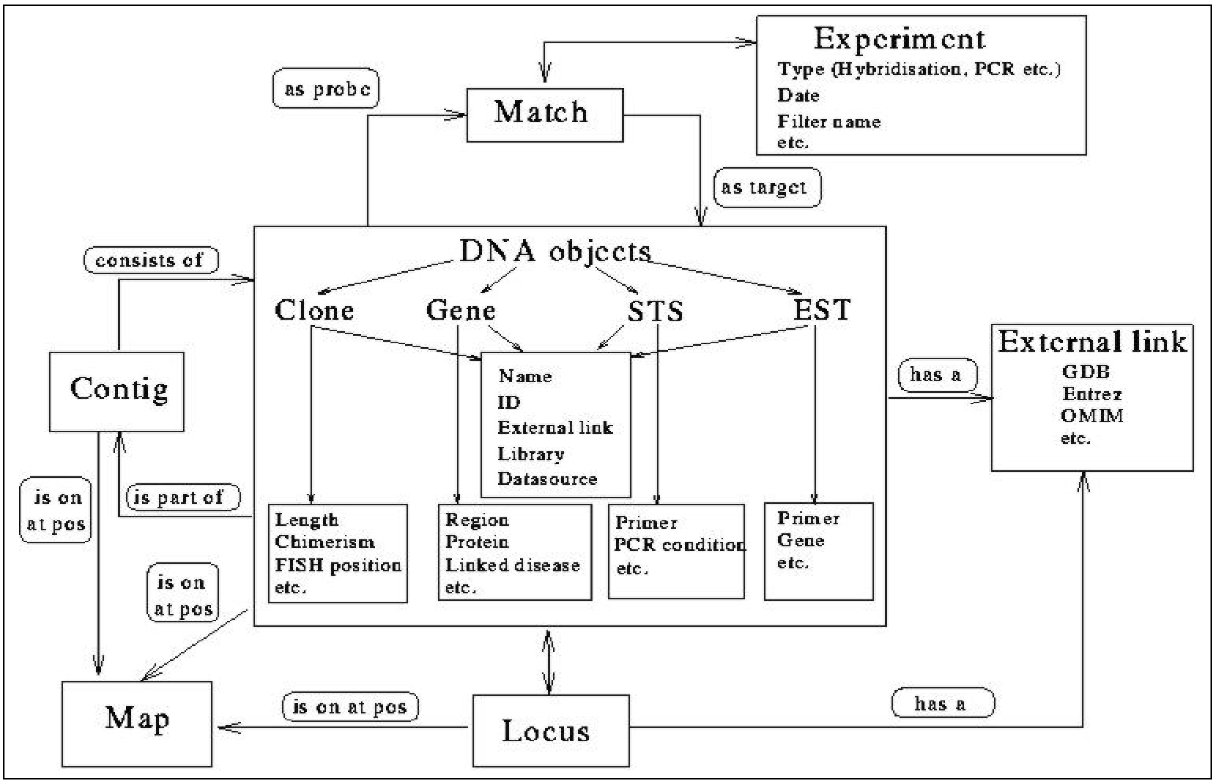


FIGURE A.21: IXDB data model ([IXDBb])

Contents. IXDB is a repository for physical mapping of the human X chromosome. Hypertext links are provide to several databases such as GDB (cf. A.35), GenBank (cf. A.36), EMBL (cf. A.28), OMIM (cf. A.72), or dbEST (cf. A.20).

Data modelling, data storage, and data acquisition. IXDB is build upon a relational model and implemented using the relational DBMS Oracle.

The data schema (Figure A.21) is centred around genomic objects in a variety of classes and consists of approximately 40 tables. Several types of information are attached to each of these classes, such as class-specific annotations, experimental results, typed relations to other objects, map positions, etc. Maps are modelled recursively so that sub-maps (in general different research groups focus on different regions of interest) can be stored and a chromosome-wide map can be constructed out of these sub-maps. The schema is highly flexible and allows new links and new object types to be established instantly.

There is no underlying ontology but to meet the frequently occurring problem, that an object is given several different names, IXDB uses synonyms. Synonyms give different names to an object, all of which can be used in queries.

For some DNA objects, however, different names describe different versions of one original unique object. A special relationship represents this situation which is called aliases. Each alias correspond to a different entry in the database, although each alias within a group is likely yield the same results, since each is likely to contain similar biological results. Therefore, queries for any one alias always retrieve a report on all aliases.

Data can be kept confidential to a user or a group of users and private views are provided for each group via a standard authentication procedure.

The original data are the result of a strict integration of data stemming from approximately 25 different chromosome scale projects. New data are acquired from other databases, the literature and the community (data submission) and added mostly automated using a special exchange format and adapted software. The software detects potential redundancy and rejects semantically inconsistent data.

Data retrieval and query answering. The WWW interface of IXDB provides a fixed-form query interface. Queries can be performed by name, keyword, external identification number (e.g. EMBL accession number), and by map position. The result will be a detailed IXDB object report on the appropriate object which contains all information stored in IXDB. The user can navigate using hypertext links to connected objects, related data, or other databases.

Sometimes several objects in IXDB overlap with the queried object. In this case, for each overlapping object the relation to the queried object is shown. The relation can be a direct connection or an overlap via an intermediate object.

There is no *ad-hoc* query interface.

Maps can be either viewed as an HTML table or viewed and compared using a graphical map display tool called ”derBrowser”, which is implemented as a Java applet. Maps can be browsed to retrieve single data items.

There is no flat file release.

Size. IXDB currently stores about 100 000 different DNA objects (excluding aliases) and 23 maps.

Literature on IXDB:

[IXDBa] `http://ixdb.mpimg-berlin-dahlem.mpg.de`

[IXDBb] U. Leseret al. IXDB, an X Chromosome Integrated Database (update). *Nucleic Acids Research*, 28(1):123-127, 2000.

A.52 KEGG: Kyoto Encyclopedia of Genes and Genomes

Provider and access. The KEGG databases are daily updated and made available through the web sites of GenomeNet in Kyoto/Japan at: `http://star.scl.genome.ad.jp/kegg/`.

Contents. The Kyoto Encyclopedia of Genes and Genomes (KEGG) Project has several objectives (see [KEGGc]):

- to computerise all aspects of cellular functions in terms of the pathway of interacting molecules or genes,

- to maintain gene catalogs for all organisms and link each gene product to a pathway component,
- to organise a database of all chemical compounds in the cell and link each compound to a pathway compound, and
- to develop computational technologies for pathway comparison, reconstruction, and analysis.

The KEGG information resource consists of three different databases:

- GENES for genomic information with a collection of gene catalogs with annotation,
- PATHWAY for higher order functional information with graphical representation of metabolism, membrane transport, etc., and
- LIGAND for information about chemical compounds involved in the cellular reactions. LIGAND is described separately (cf. A.55).

The different databases are linked via the so called DBGET/LinkDB system (cf. Chapter 4.3.2) and are integrated with several other databases, such as (among others) GenBank (cf. A.36), EMBL (cf. A.28), SWISS-PROT (cf. A.103), and PDB (cf. A.75).

Data modelling, data storage, and data acquisition. KEGG consists of the following five types of data:

- Pathway maps, represented by manually drawn diagrams
- Ortholog group tables, represented by HTML tables
- Molecular catalogs, represented b HTML tables or hierarchical texts
- Genome maps, represented by Java graphics
- Gene catalogs, represented by hierarchical texts

The DBGET/LinkDB is a home-made DBMS written in C++ that integrates these data with data from other external databases (cf. 4.3.2).

The connections between different entries of different databases are modelled using binary relations. These binary relations are stored in the LinkDB database. An entry in the web of molecular biology databases is uniquely identified by the combination of the database name and the entry name (mostly an accession number):

$$database : entry$$

and the relation between two entries within and across databases is represented by the binary relation:

$$database1 : entry1 \rightarrow database2 : entry2$$

This concept has been extended to include gene catalogs in KEGG. Any gene (or gen product) is uniquely identified by the combination of the organism and the gene name:

$$organism : gene$$

and the relation between two genes (or gene products) is represented accordingly.

Internally, data in KEGG are classified and represented as follows:

- **Entity**: a molecule or a gene.
- **Binary relation**: a relation between two entities.
- **Network**: a graph formed from a set of related entities. There are four different classes of Networks:
  - **Genome**: physical closeness of genes (stored as Java graphics).
  - **Neighbour**: similarity of molecules or genes (e.g. sequence similarity; stored as HTML tables).
  - **Hierarchy**: hierarchical classification of molecules or genes (stored as hierarchical text).
  - **Pathways**: biochemical (reference) pathway or genetic (reference) pathway (stored as GIF images).

A database in DBGET is simply considered as a collection of entries, which may be stored in a single file or multiple files. The DBGET definition of flat files includes text files and other multimedia files such as GIF files for the pathway diagrams.

The KEGG system integrates all this kind of data on the basis of the LinkDB entries. The binary relations in LinkDB are useful for comparing and computing pathways, genomes, hierarchies, and neighbours (similar sequences). For example a pathway can be computed from a given list of enzymes by deduction from binary relations of substrates and products. DBGET/LinkDB provide tools to search, compare, and handle these different representations, e.g. CGI scripts perform searching and colouring GIF image maps, and Java applets perform searching, comparing, and handling genome maps.

Data that are added to GENES are generally acquired from GenBank. The annotations of the entries are re-examined by KEGG, thus GENES was developed as a relational database using Sybase as DBMS, in order to support the procedure of data acquisition and data updates.

As mentioned above, DBGET/LinkDB is a system that integrates several flat file databases (cf. Chapter 4.3.2 for more details).

Data retrieval and query answering. The major types of the data in the KEGG system can be searched by several properties. These properties are gene/molecule name, EC number, compound number, sequence similarity, and key words. All query interfaces are fixed-form interfaces, there are no possibilities to execute *ad-hoc* queries against the KEGG system.

Several tools enable the user to browse through pathway diagrams and annotated data. E.g. reference pathway diagrams represent all known possible pathways, but in several organisms there may be only parts of a reference pathway. The user can query these parts (e.g. the parts of the lipid metabolism that occurs in yeast) and the KEGG system will visualise that by colouring the reactions which were queried. Standard analysis tools (such as BLAST [4], FASTA [56], RasMol [60], etc.) are also accessible at the GenomeNet home page.

All flat-files containing the entire data in the KEGG system (including local copies of the integrated databases such as GenBank, SWISS-PROT, DDBJ, EMBL, etc.) can be downloaded via FTP at: `ftp://kegg.genome.ad.jp/mirror/`

Size. In Spring 2000 the whole KEGG system, including GENES, PATHWAYS, LIGAND and LinkDB databases, consists of over 125 000 entries. The (reference) metabolic pathways are represented by approximately 90 diagrams.

Literature on KEGG:

[KEGGa] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27-30, 2000.

[KEGGb] `http://star.scl.genome.ad.jp/kegg/`

[KEGGc] M. Kanehisa. KEGG – From Genes to Biochemical Pathways. 1998. Available at: `http://star.scl.genome.ad.jp/kegg/kegg1.html`

[KEGGd] W. Fujibuchi et al. DBGET/LinkDB: an Integrated Database Retrieval System [26]. Available at: `http://star.scl.genome.ad.jp/kegg/kegg1.html`

A.53 KinMutBase

Provider and access. KinMutBase is maintained at the University of Tampere, Finland in collaboration with the University of Helsinki and the University of Turku. It can be accessed at: `http://www.uta.fi/imt/bioinfo/KinMutBase/`

Contents. KinMutBase is a registry of mutations in human protein kinases related to disorders. Entries in the database contain annotated information about the protein kinases.

Hypertext links to external databases such as EMBL (cf. A.28) are provided.

Data modelling, data storage, and data acquisition. There is no information available about the implementation and the underlying data model of KinMutBase. We assume that KinMutBase is a flat file database.

Data are acquired from the research community. Researchers can use a special submission tool.

Data retrieval and query answering. The KinMutBase data is classified several hierarchies that can be browsed. There are no query interfaces at all. The data can also not be downloaded via FTP. The only possibility to retrieve data from KinMutBase is thus to browse the database.

Size. No information available.

Literature on KinMutBase:

[KMBa] K. Stenberg, P. Riikonen, and M. Vihinen. KinMutBase, a database of human disease-causing protein kinase mutations. *Nucleic Acids Research*, 28(1):369-371, 2000.

[KMBb] `http://www.ncbi.nlm.nih.gov/COG/COGhelp.html`

A.54 KMDB: The Keio Mutation Database

Provider and access. The Keio Mutation Database (KMDB) is maintained and provided by the Keio University of Medicine at Tokyo/Japan. It can be accessed through: `http://mutview.dmb.med.keio.ac.jp`

Contents. KMDB stores information about mutations in human disease-causing genes. It provides hypertext links to other databases like OMIM (cf. A.72), GDB (cf. A.35), and HGMD (cf. A.42).

The database is the central system to integrate locus specific databases provided at the same institute, including: KMheartDB (Keio Mutation heart Database), KMearDB, KMbrainDB, and KMcancerDB.

Data modelling, data storage, and data acquisition. Each data item contains a gene and annotated information. These data are stored in a set of hierarchical tables according to the format defined for the "MutationView" software that integrates the different databases. MutationView has two different versions, JAVA and HTML. The JAVA version provides an interactive user interface, while the HTML version is convenient for downloading the data. There is no information on the format definition of MutationView. There is also no information about data acquisition.

Data retrieval and query answering. Data can be accessed through the MutationView interface. It provides no query interfaces but data can be browsed. The user has to register first, but the access is free of charge.

In addition, the MutationView software and the entire database can be downloaded via FTP.

Size. No information available.

Literature on KMDB:

[KMDBa] J. Murvai et al. Keio Mutation Database (KMDB) for Human Disease Gene Mutations. *Nucleic Acids Research*, 28(1):364-368, 2000.

[KMDBb] `http://mutview.dmb.med.keio.ac.jp`

A.55 LIGAND

Provider and access. LIGAND is part of the KEGG system (cf. A.52). It is maintained by the Kyoto University, Kyoto/Japan and provided at the GenomeNet homepage at: `http://star.scl.genome.ad.jp/dbget/ligand.html`

Contents. LIGAND consists of three sections:

- ENZYME for information of enzyme molecules and their reactions
- COMPOUND for information of metabolites and other chemical compounds
- REACTION for the collection of subsrate-product relations

LIGAND supports the reconstruction of metabolic pathways in KEGG (cf. A.52). It provides hypertext links to other databases.

Data modelling, data storage, and data acquisition. LIGAND is constructed as a flat file database. The data file format of each section is similar to those of GenBank (cf. A.36) and PIR (cf. A.79) flat files. A fixed number of columns are assigned to specify eah entry. See [LIGb] for detailed information about the structure of each flat file entry.

Data are acquired from the scientific literature and the KEGG database.

Data retrieval and query answering. LIGAND provides a fixed-form query interface and the possibility to browse through several enzyme classification hierarchies. Data can also be downloaded as flat files via FTP. There is no *ad-hoc* query interface.

Additional the database can be searched through the DBGet/LinkDB system (cf. Chapter 4.3.2).

Size. The current number of entries in LIGAND is available at the GenomeNet homepage. The flat files require about 5 MB (Winter 2000).

Literature on LIGAND:

[LIGa] S. Goto, T. Nishioka, and M. Kanehisa. LIGAND: Chemical Database of Enzyme Reactions. *Nucleic Acids Research*, 28(1):380-382, 2000.

[LIGb] `http://star.scl.genome.ad.jp/dbget-bin/show_man?ligand`

A.56 MAGEST: Maboya Gene Expression Patterns and Sequence Tags

Provider and access. The database for maboya gene expression patterns and sequence tags (MAGEST), is maintained by GenomeNet at the University of Kyoto. It can be accessed at: `http://star.scl.genome.ad.jp/magest`

Contents. MAGEST is a database for newly identified maternal cDNA sequences and the according expression data of the organism *Halocynthia roretzi*. The database is interconnected with several other databases via the DBGet/LinkDB system at GenomeNet (cf. Chapter 4.3.2).

Data modelling, data storage, and data acquisition. The data are stored and managed by the relational DBMS Sybase. There is no information about the underlying schema.

Data are derived from other databases, such as – among others – GenBank (cf. A.36), SWISS-PROT (cf. A.103), and PIR (cf. A.79).

Data retrieval and query answering. The database can be searched by a fixed-form query interface. Search can be performed using keywords or specifying an entry identifier. Additionally MAGEST can be searched for similarity by a reference sequence. The gene expression data is classified by the database staff and can be searched by using an arbitrary combination of expression patterns. The classified data cannot be browsed.

There is no *ad-hoc* query interface. The data can not be retrieved as flat files.

Size. No information available.

Literature on MAGEST:

[MAGa] T. Kawashima et al. MAGEST: Maboya Gene Expression Patterns and Sequence Tags. *Nucleic Acids Research*, 28(1):133-135, 2000.

[MAGb] `http://www.cme.msu.edu/RDP/docs/documentation.html`

A.57 MaizeDB

Provider and access. MaizeDB is accessible at: `http://www.agron.missouri.edu/`

Contents. MaizeDB provides information about the maize genome and its expression. It seems that there are currently no interconnections with other databases.

Data modelling, data storage, and data acquisition. MaizeDB is implemented using the relational DBMS Sybase. Entity-Relationship diagrams for all classes of the database schema can be viewed at [MaiDBb].

An ACEDB version is also available. Its schema consists of 18 classes, whose model can be browsed at [maiDBc].

It is not really sure where the data for MaizeDB are acquired.

Data retrieval and query answering. MaizeDB can be accessed through the so called Object Protocol Management (OPM), which should not be mistaked for the model to define a database schema (OPM, cf. Chapter 3.3.3). This OPM is a Java-based interface, provided by Gene Logic[1], that works best on local PCs. The interface provides three ways of searching:

- full text search
- browsing through the classes hierarchically
- query for a name in a special class

An *ad-hoc* query interface is under development for the future.

There is no flat file release for downloading the data.

Size. No information available.

Literature on MaizeDB:

[MaiDBa] `http://www.agron.missouri.edu/schema.html`

[MaiDBb] `http://grain.jouy.inra.fr/cgi-bin/webace/webace?db=maizedb`

A.58 MDDB: The Metabolic Disease Database

Provider and access. The Metabolic Disease Database (MDDB) is developed and maintained at the Otto-von-Guericke-University Magdeburg, Germany. It can be accessed at: `http://edradour.cs.uni-magdeburg.de/iti_bm/mddb/` (download of the software), or at: `http://www-bm.cs.uni-magdeburg.de/iti_bm/bmbf/mdcave.html` (web interface).

Contents. The MDDB provides data on medical, genetic and biochemical aspects of metabolic diseases. These database is designed for medical research as well as biochemical research.

There is no information about links to external databases.

Data modelling, data storage, and data acquisition. The database relies on a relational model. The DBMS Oracle is used to manage the data in MDDB. There is no information available about details of the underlying relational schema.

There is no information available about data acquisition in MDDB.

[1] `www.genelogic.com`

Data retrieval and query answering. The web interface called MD-Cave requires an account (most likely free off charge) and allows direct access to the MDDB trough the web. MD-Cave is a Java Applet and is connected via JDBC to the Oracle DBMS server.

There is no information available about the capabilities of data retrieval and the query interfaces.

The software and contents of MDDB are available within a prototype package. The software and data package can be downloaded (see URL above) for local installation.

Size. No information available.

Literature on MDDB:

[MDDBa] `http://www-bm.cs.uni-magdeburg.de/iti_bm/mddb/`

A.59 MEDLINE

MEDLINE seems to be re-implemented at the National Center for Biotechnology Information (NCBI) in PubMed (cf. A.90)

A.60 MEROPS

Provider and access. The MEROPS database is maintained at the Brabham Institute near Cambridge, UK. It can be accessed at: `http://www.merops.co.uk/merops/merops.htm`

Contents. The MEROPS database provides a catalogue and structure-based classification of proteases (termed "peptidases" within the database), together with additional information about them. It provides hypertext links to SWISS-PROT (cf. A.103) and FlyBase (cf. A.34).

Data modelling, data storage, and data acquisition. The MEROPS database is most likely a flat file database consisting of explicitly structured entries. There is no information about the structure of the database.

Data are primarily acquired from external databases.

Data retrieval and query answering. MEROPS can be accessed via several fixed-form query interfaces (keyword search against a search index). There is no *ad-hoc* query interface. The data are classified in several hierarchies that can be browsed for data retrieval.

In addition, the database can be downloaded as flat file via FTP.

Size. No information available.

Literature on MEROPS:

[MEROa] N. Rawlings and A. Barret. MEROPS: The Peptidase Database. *Nucleic Acids Research*, 28(1):323-325, 2000.

[MEROb] `http://www.merops.co.uk/merops/merops.htm`

A.61 MetaCyc

see EcoCyc (A.25 at page 141)

A.62 MGD: The Mouse Genome Database

Provider and access. The Mouse Genome Database (MGD) is maintained by The Jackson Laboratory at Bar Harbor/USA and accessible together with GXD (cf. A.39) and MTB (cf. A.70) at: `http://www.informatics.jax.org/`

Contents. The MGD is a public database of mouse genomic, genetic, and phenotypic information. MGD provides hypertext links to several other databases. See [MGDb] for a comprehensive list of cross-linked databases.

Data modelling, data storage, and data acquisition. MGD is implemented using the relational DBMS Sybase. There is no information about the underlying schema of the database. Probably, if one registers for direct access via SQL-queries, detailed information on the database schema will be provided.

In collaboration with SGD (cf. A.100) and FlyBase (cf. A.34), MGD is developing a biological ontology to use standardised vocabulary to annotate genes or gene products.

Data are acquired from other databases and from submissions of the scientific community. The database is updated daily.

Data retrieval and query answering. Data from MGD are accessible via several fixed-form query interfaces or indirectly by browsing through clickable maps.

Users may also access MGD directly via an *ad-hoc* query interface, expressing their query in SQL. But this requires registration first. Registrated users get support via e-mail and phone.

Flat file versions of MGD are available for download at MGD's homepage or at several mirror sites around the world (see [MGDa]).

Standard data analysis tools are with the database.

Size. No information available.

Literature on MGD:

[MGDa] J. Blake et al. The Mouse Genome Database (MGD): Expanding Genetic and Genomic Resources for the Laboratory Mouse. *Nucleic Acids Research*, 28(1):94-99, 2000.

[MGDb] `http://www.informatics.jax.org/userdocs/overview.shtml`

A.63 MIPS: The Munich Information Center for Protein Sequences

Provider and access. The Munich Information Center for Protein Sequences (MIPS-GSF) near Munich/Germany provides several high quality curated genome databases. All databases can be accessed at: `http://www.mips.biochem.mpg.de`

Contents. Databases provided at MIPS are:

- **MATDB:** The MIPS *Arabidopsis thaliana* Database (MATDB) stores comprehensive information about the sequences of the *Arabidopsis thaliana* genome. Sequence data are annotated with biological information.
- **MNCDB:** The MIPS *Neurospora crassa* Database (MNCDB) is the result of the genome analysis of two chromosomes of *Neurospora crassa*. MIPS is responsible for analysing the data, gene modelling and annotating predicted proteins and other genetic features of chromosome II and V within the German *N. crassa* sequencing project.
- **MYGD:** The MIPS Yeast Genome Database (MYGD) presents a comprehensive database which summarises the current knowledge regarding the more than 6000 ORFs encoded by the Yeast Genome.
- **HIB:** The Human Information Base (HIB) is a database of automatically annotated human gene clusters including a functional classification of human proteins based on systematic homology and pattern analysis.
- **Database of the complete cDNA sequences from the DHGP:** is a result of efforts within the German Human Genome Project (DHGP). All EST and complete cDNA data are stored in this database.
- **ProtFam:** The Protein Sequence Homology Database (ProtFam) is a curated database of homology clusters (protein super-families, protein families and homology domains). Classification results are directly copied to the entries of the PIR-International Protein Sequence Database (PSD better known as PIR).
- **PEDANT:** PEDANT is a standalone program package that performs a complete structural and functional characterisation of protein sequence sets using a suite of highly sophisticated bioinformatics tools.

Data items are hypertext linked within MIPS and to other data sources.

Data modelling, data storage, and data acquisition. The Database of the complete cDNA sequences from the DHGP is stored in an object database. PEDANT is implemented using the freely available relational DBMS MySQL. There is no information about the implementation of the other databases at MIPS nor on the modeling of all MIPS resources.

In general, new data are acquired by MIPS from collaborating research groups and submissions from the community.

Data retrieval and query answering. All databases at MIPS are accessible via several fixed-form query interfaces. There is no *ad-hoc* query interface available. The data can be browsed.

There is no flat file distribution available.

Size. No information available.

Literature on MIPS:

[MIPSa] H.W. Mewes et al. MIPS: A Database for Genomes and Protein Sequences. *Nucleic Acids Research*, 28(1):37-40, 2000.

[MIPSb] `http://www.mips.biochem.mpg.de`

A.64 MitBASE

Provider and access. The database is maintained by a collaboration between the European Bioinformatics Institute (EBI) at Cambridge/UK, the University of Bari/Italy, and several other academic institutes. MitBASE is accessible at: `http://www3.ebi.ac.uk/Research/Mitbase/mitbase.pl`

Contents. MitBASE is an integrated database of mitochondrial DNA data. It collects under a single interface databases for Plant, Vertebrate, Invertebrate, Human, Protist and Fungal mitochondrial DNA.

MitBASE is a collection of specialised ”sub-databases”. There is no information about hypertext links provided by MitBASE.

Data modelling, data storage, and data acquisition. The whole database is centrally structured under an ORACLE DBMS at EBI. For each sub-database an ORACLE database structure has been designed and implemented. Moreover, each sub-database is locally managed by commercial DBMSs such as MS Access and FileMaker Pro, or simple editor systems working on flat file data.

The sub-databases currently include (see [MitBa] for further information):

- Human MitBASE database
- Fungal MitBASE database
- MitBASE Pilot database
- Plant and Algae MitBASE database
- Protist MitBASE database
- Vertebrate database
- Invertebrate MitBASE database

Data are acquired from the literature and external databases such as EMBL (cf. A.28) and GenBank (cf. A.36). Data from personal communications are also added. Individual researchers are able to submit updates via a web-based submission form.

Data retrieval and query answering. The integrated MitBASE database can be queried at the EBI web site through different approaches or can released in a flat file format suitable for the Sequence Retrieval System (SRS, cf. Chapter 4.3.7).

The approaches to retrieve data from MitBASE are:

- to browse a hierarchical structure of the keywords related to mitochondrial gene names and regulatory regions
- to execute fixed-form queries via a dedicated interface
- to use the SRS at EBI for searching

There is no interface to express and execute *ad-hoc* queries.

A list of the mitochondrial translation tables specific to each organism is also provided to help researchers translating nucleotide sequences into amino acid sequences.

There is no flat file distribution offered for download.

Size. Currently MitBASE contains approximately 14 000 entries. Detailed information about the current size is provided at [MitBb].

Literature on MitBASE:

[MitBa] M. Attimonelli et al. MitBASE: a comprehensive and integrated mtDNA database: the present status. *Nucleic Acids Research*, 28(1):148-152, 2000.

[MitBb] `http://www3.ebi.ac.uk/Research/Mitbase/mitbase.pl`

A.65 MitoNuc/MitoAln

Provider and access. MitoNuc and MitoAln are provided by a collaboration of the University of Bari, the University of Milan, and the C.N.R. at Bari (Italy). Both databases can be accessed at: `http://bighost.area.ba.cnr.it/srs/`

Contents. MitoNuc and MitoAln are two related databases providing information on genes that code for mitochondrial proteins and are encoded in the nuclear DNA. MitoNuc is the resource that contains all annotated nucleotide sequences of the according genes. MitoAln contains gene entries structured in homology families and relevant nucleotide and protein multi-alignments.

Hypertext links to external databases such as among others EMBL (cf. A.28) and SWISS-PROT (cf. A.103) are provided. Both MitoAln and MitoNuc databases are cross-linked with each other.

Data modelling, data storage, and data acquisition. MitoNuc and MitoAln are both flat file databases. Each database consists of entries that are explicitly structured by line types (search indexes). MitoNuc and MitoAln entry flat file structures are presented in [MITOa].

Data are acquired from other databases (primarily from EMBL) and the scientific literature. Sequence alignments in MitoAln are produced automatically and are optimised manually.

Data retrieval and query answering. MitoNuc and MitoAln can only be accessed via the Sequence Retrieval System (SRS, cf. Chapter 4.3.7). A local SRS server is installed at the MitoNuc/MitoAln homepage.

There is no further interface to both databases. The files cannot be downloaded via FTP.

Size. No information available.

Literature on MitoNuc/MitoAln:

[MITOa] G. Pesole et al. MitoNuc and MitoAln: two related databases of nuclear genes coding for mitochondrial proteins. *Nucleic Acids Research*, 28(1):163-165, 2000.

[MITOb] `http://bio-www.ba.cnr.it:8000/BioWWW/`

A.66 MITOP: The Mitochondrial Proteome Database

Provider and access. The Mitochondrial Proteome Database (MITOP) is build upon a collaboration of several academic institutes in Munich, Germany and is maintained by the Munich Information Centre for Protein Sequences (MIPS) at Munich, Germany. MITOP can be accessed at: `http://www.mips.biochem.mpg.de/proj/medgen/mitop`

Contents. MITOP is a comprehensive database for genetic and functional information on both nuclear- and mitochondrial-encoded proteins and their genes. It provides hypertext links to several other databases and the literature (see [MITOPb]).

```
>P1;CCMST - cytochrome c, testis-specific - mouse
MGDAEAGKKIFVQKCAQCHTVEKGGKHKTGPNLWGLFGRKTGQAPGFSYTDANKNKGVIW
SEETLMEYLENPKKYIPGTKMIFAGIKKKSEREDLIKYLKQATSS
>P1;CBMS - ubiquinol--cytochrome-c reductase (EC 1.10.2.2) cytochrome b - mouse
MTNMRKTHPLFKIINHSFIDLPAPSNISSWWNFGSLLGVCLMVQIITGLFLAMHYTSDTM
TAFSSVTHICRDVNYGWLIRYMHANGASMFFICLFLHVGRGLYYGSYTFMETWNIGVLLL
FAVMATAFMGYVLPWGQMSFWGATVITNLLSAIPYIGTTLVEWIWGGFSVDKATLTRFFA
FHFILPFIIAALAIVHLLFLHETGSNNPTGLNSDADKIPFHPYYTIKDILGILIMFLILM
TLVLFFPDMLGDPDNYMPANPLNTPPHIKPEWYFLFAYAILRSIPNKLGGVLALILSILI
LALMPFLHTSKQRSLMFRPITQILYWILVANLLILTWIGGQPVEHPFIIIGQLASISYFS
IILILMPISGIIEDKMLKLYP
>P1;DEMSMM - malate dehydrogenase (EC 1.1.1.37) precursor, mitochondrial - mouse
MLSALARPAGAALRRSFSTSAQNNAKVAVLGASGGIGQPLSLLLKNSPLVSRLTLYDIAH
TPGVAADLSHIETRANVKGYLGPEQLPDCLKGCDVVVIPAGVPRKPGMTRDDLFNTNATI
VATLTAACAQHCPEAMVCIIANPVNSTIPITAEVFKKHGVYNPNKIFGVTTLDIVRANTF
VAELKGLDPARVNVPVIGGHAGKTIIPLISQCTPKVDFPQDQLATLTGRIQEAGTEVVKA
KAGAGSATLSMAYAGARFVFSLVDAMNGKEGVVECSFVQSKETECTYFSTPLLLGKKGLE
KNLGIGKITPFEEKMIAEAIPELKASIKKGEDFVKNMK
>P1;JC4210 - 3-hydroxyacyl-CoA dehydrogenase (EC 1.1.1.35), short
MAFVTRQFLRSMSSSSSASAAAKKILIKHVTVIGGGLMGAGIAQVAAATGHTVVLVDQTE
DILAKSKKGIEESLKRMAKKKFTENPKAGDEFVEKTLSCLSTSTDAASVVDSTDLVVEAI
VENLKLKNELFQRLDKFAAEHTIFASNTSSLQITNIANATTRQDRFAGLHFFNPVPMMKL
VEVIKTPMTSQKTFESLVDFCKTLGKHPVSSKDTPGFIVNRLLVPYLIEAVRLHERGDAS
KEDIDTAMKLGAGYPMGPFELLDYVGLDTTKFILDGWHEMEPENPLFQPSPSMNNLVAQK
KLGKKTGEGFYKYK
>P1;I48966 - aldehyde dehydrogenase (NAD+) (EC 1.2.1.3) 2 precursor,
MLRAALTTVRRGPRLSRLLSAAATSAVPAPNHQPEVFCNQIFINNEWHDAVSRKTFPTVN
PSTGEVICQVAEGNKEDVDKAVKAARAAFQLGSPWRRMDASDRGRLLYRLADLIERDRTY
LAALETLDNGKPYVISYLVDLDMVLKCLRYYAGWADKYHGKTIPIDGDFFSYTRHEPVGV
CGQIIPWNFPLLMQAWKLGPALATGNVVVMKVAEQTPLTALYVANLIKEAGFPPGVVNIV
PGFGPTAGAAIASHEGVDKVAFTGSTEVGHLIQVAAGSSNLKRVTLELGGKSPNIIMSDA
DMDWAVEQAHFALFFNQGQCCCAGSRTFVQENVYDEFVERSVARAKSRVVGNPFDSRTEQ
GPQVDETQFKKILGYIKSGQQEGAKLLCGGGAAADRGYFIQPTVFGDVKDGMTIAKEEIF
GPVMQILKFKTIEEVVGRANDSKYGLAAAVFTKDLDKANYLSQALQAGTVWINCYDVFGA
QSPFGGYKMSGSGRELGEYGLQAYTEVKTVTVKVPQKNS
```

FIGURE A.22: First five entries of the "mouse" flat file in MITOP.

Data modelling, data storage, and data acquisition. MITOP consists of five flat files with sequence and annotation data. Each file contains information on a specific organism:

- *S. cerevisiae* (yeast)
- *M. musculus* (mouse)
- *C. elegans* (worm)
- *N. crassa* (bread mold)
- *H. sapiens* (human)

The first five entries of the file with the data concerned to the mouse is shown in Figure A.22.

Data are acquired from other databases and the literature.

Data retrieval and query answering. Data in MITOP can be retrieved in several different ways. A simple fixed-form query interface supports searching by keyword or gene names. Several entry points allow hierarchical browsing of the database by distinct topics such as protein classes, protein complexes, subcellular location, EC number, etc. The MITOP sequence files can also be downloaded via FTP.

There is no *ad-hoc* query interface.

Size. The files currently require:

S. cerevisiae (yeast)	170 KB
M. musculus (mouse)	44 KB
C. elegans (worm)	20 KB
N. crassa (bread mold	32 KB
H. sapiens (human)	118 KB
Sum	**384 KB**

Literature on MITOP:

[MITOPa] C. Scharfe et al. MITOP, the Mitochondrial Proteome Database: 2000 Update. *Nucleic Acids Research*, 28(1):37-40, 2000.

[MITOPb] `http://www.mips.biochem.mpg.de/proj/medgen/mitop/description`

A.67 MMDB: The Molecular Modeling DataBase

Provider and access. The Molecular Modeling DataBase (MMDB) is provided by the NCBI/NIH at Bethesda/USA and can be accessed through NCBI's retrieval system Entrez at: `http://www.ncbi.nlm.nih.gov:80/Structure/MMDB/mmdb.shtml`.

Contents. MMDB is a protein structure database. MMDB provides links to several other databases. Via Entrez (cf. Chapter 4.3.4, MMDB is connected to all other databases at NCBI, such as GenBank (cf. A.36), OMIM (cf. A.72), PubMed (cf. A.90), etc. There are also hypertext links to the PDB.

Data modelling, data storage, and data acquisition. MMDB is a database of ASN.1[5]-formatted records. The records are stored in files.

Data are derived from the Protein Data Bank (PDB, cf. A.75).

Data retrieval and query answering. MMDB can be accessed through Entrez (cf. Chapter 4.3.4). Entrez supports searching via a fixed-form query interface against protein names, author names, etc. In addition, results of these queries can be logically combined. Data can be downloaded via FTP.

There is no *ad-hoc* query interface, the data cannot be browsed.

Several analysis tools are accessible via the database. E.g. neighbours are identified automatically using the BLAST [4] (sequence neighbours) or the VAST (structure neighbours) algorithms. Several visualisation tools allow representing the 3D structure of the proteins.

Size. Currently MMDB contains of over 14 000 files, each requiring between approximately 5 KB and 1.5 MB. MMDB is currently updated once per month.

Literature on MMDB:

[MMDBa] Y. Wang et al. MMDB: 3D Structure Data in Entrez. *Nucleic Acids Research*, 28(1):243-245, 2000.

[MMDBb] `http://www.ncbi.nlm.nih.gov:80/Structure/MMDB/mmdb.shtml`

A.68 ModBase

Provider and access. ModBase is maintained at the Rockefeller University, New York, USA and can be accessed through a web interface at: `http://pipe.rockefeller.edu/modbase/`.

Contents. ModBase is a database of annotated comparative protein structure models. These models are generated by a computer program called MODPIPE, relying on the programs PSI-BLAST and MODELLER. The database currently stores 3D models for substantial portions of about 17 000 proteins from several organisms. ModBase contains theoretically calculated models, which may contain significant errors, not experimentally determined structures.

Hypertext links to several other databases, such as PDB (cf. A.75), CATH (cf. A.12), SWISS-PROT (cf. A.103), etc., are provided.

Data modelling, data storage, and data acquisition. The database stores structure models for segments of proteins. Each model has its atom coordinates stored in a flat file in PDB format (cf. A.75). Currently ModBase is undergoing some transitions, thus the database will be perhaps implemented next year using a standard DMBS. But it seems more likely that ModBase is just migrating to another location and don not change its flat file implementation..

A list of data source of ModBase is available at [ModBb], following the link “sequence sources”.

Data retrieval and query answering. Due to the undergoing transition of ModBase, currently only a subset of the models in the database is available at [ModBb].

ModBase is accessible via its home page at [ModBb], providing several fixed-form query interfaces. The database is searchable by several features and items such as SWISS-PROT (cf. A.103) accession numbers, ORF names, keywords, model size, etc. It is also possible to perform sequence similarity search against the model sequences using BLAST [4].

There is no *ad-hoc* query interface, no possibility to download the data via FTP, and no hierarchy to browse ModBase.

Size. No information available.

Literature on ModBase:

[ModBa] R. Sanches et al. ModBase, a Database of Annotated Comparative Protein Structure Models. *Nucleic Acids Research*, 28(1):250-253, 2000.

[ModBb] `http://pipe.rockefeller.edu/modbase/`

A.69 MPW: The metabolic Pathway Database

The Metabolic Pathway Database (MPW) was originally developed at the Institute of Theoretical and Experimental Biophysics in Moscow, Russia and is most likely now maintained at WIT (cf. A.115).

A.70 MTB: The Mouse Tumor Database

Provider and access. The Mouse Tumor Biology Database (MTB) is provided by the Mouse Genome Informatics Group of The Jackson Laboratory at Bar Harbor/USA. It is accessible via WWW together with the MGD (cf. A.62) and GXD (cf. A.39) at: `http://informatics.jax.org`

Contents. The MTB provides access to information about tumors in genetically defined mice. A major focus for database development during the past years was to increase the number and diversity of tumor histopathologic images. Certain data items are (hypertext) linked to other databases such as the MGD (cf A.62), the GXD (cf. A.39), and MEDLINE (cf. A.59).

Data modelling, data storage, and data acquisition. The MTB was originally implemented as a relational database using the relational DBMS FileMaker Pro 4. It is still a prototype. The schema consists of 36 tables and is described in more detail in [MTBb]. In the future, MTB is going to migrate to Sybase and to integrate it with other databases in the MGI Group (GXD, MGD).

Data are classified in six key information areas (tumor type, tumor frequency and latency, tumor genetics, tumor pathology, mouse strain, and reference).

Controlled vocabularies are used in MTB to ensure accuracy and completeness of responses to user queries.

Data are primarily obtained from the scientific literature and direct submissions from the research community.

Data retrieval and query answering. Access to MTB is provided using multiple web-based fixed-form query interfaces. Each form corresponds to one of the six key information classes. In addition, the database can be searched by accession ID of any object in the database.

There is no interface to express and execute *ad-hoc* queries. Flat file release for download is not provided. There is no hierarchy to browse through for data retrieval.

Size. No information available.

Literature on MTB:

[MTBa] C. Bult et al. Mouse Tumor Biology Database (MTB): enhancements and current status. *Nucleic Acids Research*, 28(1):112-114, 2000.

[MTBb] C. Bult, D. Krupke, and J. Eppig. Electronic Access to Mouse Tumor Data: the Mouse Tumor Biology Database (MTB) Project. *Nucleic Acids Research*, 27(1):99-105, 1999.

[MTBc] `http://tumor.informatics.jax.org/about\protect\unhbox\voidb@x\kern.06em\vbox{\hrulewidth.3em}mtb.html`

A.71 NDB: The Nucleic Acid Data Bank

Provider and access. The Nucleic Acid Data Bank (NDB) is provided by the Rutgers University/USA. It is accessible at: `http://ndbserver.rutgers.edu:80/`.

Contents. NDB provides annotated data on three dimensional structures of nucleic acids resulted from NMR experiments. It is cross-linked with the PDB (cf. A.75) and the BMRB (cf. A.10).

Data modelling, data storage, and data acquisition. The NDB is most likely internally managed by a relational DBMS. The data are converted into the so-called “mmCIF” file format which is part of the “Self-Defining Text Archive and Retrieval” (STAR) format [32]. The STAR format is used as data exchange format.

Data in the NDB are acquired from the literature, external databases and submissions of researchers. A special data submission tool is provided to assist biochemical researchers in data submission to NDB.

Data retrieval and query answering. NDB provides a fixed-form query interface where the user can fill in some query forms.

In addition, the NDB interface at [NDBb] provides the capability to browse through tables (these tables seemed to be the existing tables of the underlying relational model) and their attributes. To retrieve a single data item the user must fill in keywords that will be used to search against the chosen attribute of the chosen table.

There are no *ad-hoc* query interfaces nor capabilities to download the data as files.

Size. No information available.

Literature on NDB:

[NDBa] H. Berman et al. The Nucleic Acid Database: present and future. *Journal of Research of the National Institute of Standards and Technology*, 101(3):234-257

[NDBb] `http://ndbserver.rutgers.edu:80/`

A.72 OMIM: Online Mendelian in Men

Provider and access. The Online Mendelian Inheritance in Man (OMIM) database is developed at Johns Hopkins University and is maintained by the National Center for Biotechnology Information (NCBI), USA. It can be accessed via NCBI's integrated retrieval system Entrez (cf. Chapter 4.3.4).

Contents. OMIM is a catalog of human genes and genetic disorders. It provides links to all databases at the NCBI via Entrez (cf. Chapter 4.3.4) and several external sources.

Data modelling, data storage, and data acquisition. There is no information available about the data modelling and management. In fact, there is a flat file release of OMIM (see below) but that does not mean that OMIM is stored internally at the NCBI as a flat file database. More likely, the data is stored as ANS.1 [5] structured files.

Data are probably acquired from the research community (especially from the Johns Hopkins University).

Data retrieval and query answering. OMIM provides several complex fixed form query interfaces. Search strings may consist of a keyword or several keywords combined with boolean operators (see [OMIMb] for details).

The entire database can be downloaded as flat file(s) via FTP at: `ftp://ncbi.nlm.nih.gov/repository/OMIM/`

There are no *ad-hoc* query interface nor a browsable hierarchy for data retrieval.

Size. The compressed flat file of OMIM requires currently about 30 MB.

Literature on OMIM:

[OMIMa] D. Wheeler et al. Database Resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 28(1):10-14, 2000.

[OMIMb] `http://www.ncbi.nlm.nih.gov:80/entrez/Omim/omimhelp.html`

A.73 ooTFD: The object-oriented Transcription Factors Database

Provider and access. The object-oriented Transcription Factors Database (ooTFD) is maintained by the Institute for Transcriptional Informatics at Pittsburgh/USA. It can be accessed at: `http://www.ifti.org/`.

Contents. ooTFD contains information about transcription factor binding sites, as well as composite relationships within transcription factors. It provides hypertext links to major sequence databases such as GenBank (cf. A.36), EMBL (cf. A.28), SWISS-PROT (cf. A.103) and PSD/PIR. (cf. A.79)

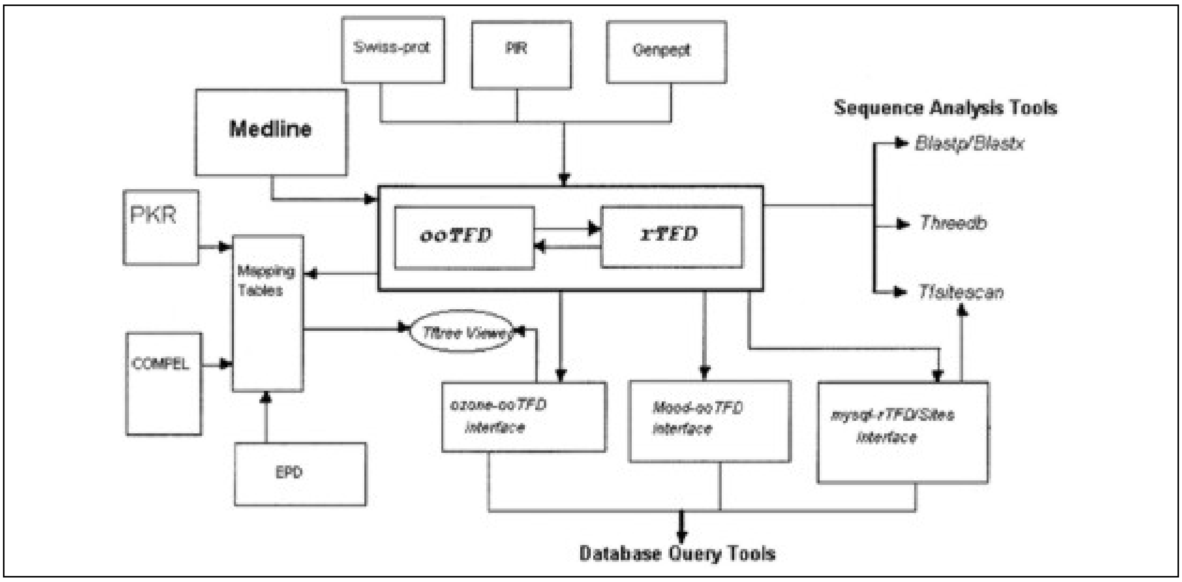


FIGURE A.23: Data flow between ooTFD, its tools, and other resources (taken from [ooTFDa])

Data modelling, data storage, and data acquisition. The ooTFD is based on data structures derived from the tables of the original database, now referred to as rTFD (relational Transcription Factors Database). ooTFD has been implemented in a number of object DBMSs including ROL[2] (Rule-based

[2] `http://cs.uregina.ca/~mliu/ROL/rol-release22.html`

Object Language), MOOD[3] (Materials object-oriented database), and the pure java object database ozone[4].

The most importatnt parts of the underlying schema are presented in [TDFa]. The flow of information between this database, related data resources, and query and analysis tools is shown in Figure A.23

Data are acquired from the scientific literature and external databases such as PSD/PIR (cf. A.79) and SWISS-PROT (cf. A.103).

Data retrieval and query answering. The two categories of retrieval tools and services of ooTFD, namely sequence analysis queries and database queries, are summarised in the following [TFDa and TFDb]:

- a tool called ”Tfsitescann”: Sites analyses can be performed, a service which constructs an image-map in association with sequence analysis results which is linked to individual Sites entries.
- a tool called ”Tfdaa/BLASTP”: Protein sequence analyses can be performed against the Polypeptides/Domains dataset (tfdaa) through a standard BLASTP (part of BLAST tools [4]) analysis.
- a tool called ”TF-Advisor”: A simple ”knowledge-based”web tool which is available to a new user for selecting the most suitable approach to performing a particular query.
- a tool called ”ooTfd query tools”: Object database queries are currently possible through an interface to the ozone implementation of ooTFD, and these generally provide links to the individual database entries.

In addition the data can be downloaded via FTP at several mirror sites.

There is no *ad-hoc* query interface nor a possibility to browse the data.

Size. Currently the zipped flat file of the MOOD database requires 3089 KB and the zipped file of ozone requires 235 KB.

Literature on ooTFD:

[TFDa] D. Ghosh. Object-oriented Transcription Factors Database (ooTFD). *Nucleic Acids Research*, 28(1):308-310, 2000.

[TFDb] `http://www.ifti.org/`

[3] `http://mood.mech.hi-tech.ac.jp/home.html`
[4] `http://www.softwarebuero.de/ozone-eng.html`

A.74 ORDB: The Olfactory Receptor Database

Provider and access. The Olfactory Receptor Database (ORDB) is maintained at the Yale University School of Medicine, New Haven/USA. ORDB is a member of a consortium that provides several databases with information about special proteins [ORDBa]. It is accessible at: `http://ycmi.med.yale.edu/senselab/ordb/`

Contents. ORDB is a database containing properties and sequences of special proteins. Hypertext links to several other databases are provided.

Data modelling, data storage, and data acquisition. The data schema is based upon a model called "EAV/CR" (Entity-Attribute-Value with Classes and Relationships). EAV/CR can be seen as an object-relational model. It extends the basic principle of row modelling to allow complex data values (classes) and the explicit representation of relationships in the database [ORDBa]. There is no information about the concrete schema and the DBMS.

New data seem to be acquired only from collaborating laboratories.

Data retrieval and query answering. ORDB provides a fixed-form query interface for keyword search. The database can also be browsed. An interface for a BLAST [4] sequence similarity search is also available.

There is no *ad-hoc* interface and no flat file release for download.

Although there is no such information provided, users seemingly have to register first before retrieving data from ORDB.

Size. ORDB currently contains of about 1100 objects. There is no further information available.

Literature on ORDB:

[ORDBa] E. Skoufos et al. Olfactory Receptor Database: a Sensory Chemoreceptor Resource. *Nucleic Acids Research*, 28(1):341-342, 2000.

[ORDBb] `http://ycmi.med.yale.edu/senselab/ORDB/`

A.75 PDB: The Protein Data Bank

Provider and access. The Protein Data Bank was established at Brookhaven National Laboratories in 1971. In October 1998, the management of the database became the responsibility of the Research Collaboratory for Structural Bioinformatics (RCSB). The PDB is accessible at: `http://www.rcsb.org/pdb/`

Contents. The PDB is the single worldwide archive of structural data of biological macromolecules. PDB entries are linked to MEDLINE (cf. A.59). The RSCB web page provides a collection of useful links to other databases.

Data modelling, data storage, and data acquisition. Currently the PDB is based upon five major components (see [PDBa]):

- the core database provides the central physical storage for the primary data in PDB. It is managed by the relational DBMS Sybase.
- a database contains indexed objects with native and derived properties. This database is based upon a model called Property Object Model (POM).
- the Biological Macromolecule Crystallisation Database (BMCD) contains three general categories of literature derived information (macromolecular, crystal, and summary data). It is managed by the relational DBMS Sybase.
- the Netscape LDAP server is used to index the textual content of the PDB in a structured format. It also provides support for keyword searches.
- the final (curated) data are stored in files

In the following we will take a look on the final data stored as flat files: Each PDB entry is stored in an own flat file. The format of an entry is defined in the Guide to Authors v2.1 (draft) provided by the PDB at [PDBc]. The database currently contains entries in at least four distinct formats, v1.0, v2.0, v2.1, and v2.2. There is significant undocumented variation within a given format that makes e.g. consistent parsing difficult [PDBb].

There is an underlying ontology of about 1700 terms that define the macromolecular structure and the crystallographic experiment. This ontology, which is called “macromolecular Crystallographic Information File” (mmCIF) dictionary, is only developed for data derived from crystallographic experiments. The mmCIF dictionary defines also an alternative file format the so-called STAR format [32]. An ontology for data derived from NMR is currently under construction. The final curated files are stored as ASCII files in PDB and mmCIF formats in the FTP archive. PDB files in XML format are planned as future developments.

Data are acquired from the research community by submission. Data may be submitted via email or a tool called ”AutoDep Input Tool” (ADIT). ADIT is also built upon the mmCIF ontology. Several checks during data acquisition ensure correctness. Details on the data acquisition procedures can be obtained from [PDBa].

Data retrieval and query answering. The PDB can be searched via different fixed-form query interfaces. The database can be searched by PDB ID or by keywords. A third interface provides a fill-in form to query several fields of the PDB.

The data is also distributed to the community in the following ways:

- flat file download via FTP (updated weekly) from RCSB.
- flat file download via FTP from other mirror sites.

- CD-ROM (quarterly released)

There is no *ad-hoc* query interface nor a possibility to browse the data.

Size. The PDB currently stores 13672 structures. A sample file containing one entry e.g. requires about 200 KB. But there is no information about the average size of the entries nor about the size of the entire database.

Literature on PDB:

[PDBa] H. Berman et al. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235-242, 2000.

[PDBb] H. Weissig and P. Bourne. An Analysis of the Protein Data Bank in Search of Temporal and Global Trends.

[PDBc] `http://www.rcsb.org/pdb/info.html`

A.76 PEDANT

see MIPS (A.63 at page 186)

A.77 PEDB: The Prostate Expression Database

Provider and access. The Prostate Expression Database (PEDB) is maintained at Seattle/USA. It is freely accessible at: `http://www.pedb.org/`

Contents. PEDB stores gene expression information derived from human prostate. It provides hypertext links to other relevant WWW resources (databases and tools).

Data modelling, data storage, and data acquisition. PEDB is implemented upon a relational DBMS. Further information about the schema and data flow of PEDB is presented at [PEDBb].

Data are acquired from other databases and the scientific literature.

Data retrieval and query answering. PEDB can be queried by BLAST [4] similarity search or via a fixed-form query interface. A data analysis tool called ”Virtual Expression Analysis Tool” (VEAT), written in Java, provides the ability to browse expression data of different species.

For access to PEDB, the user has to register at first (free of charge).

Most likely there are no query interfaces to access PEDB nor a flat file release for download via FTP.

Size. No information available.

Literature on PEDB:

[PEDBa] P. Nelson et al. The Prostate Expression Database (PEDB): Status and Enhancements 2000. *Nucleic Acids Research*, 28(1):212-214, 2000.

[PEDBb] `http://www.pedb.org/OVERVIEW/`

A.78 Pfam: The Protein Families Database

Provider and access. The Pfam database is maintained at the Sanger Centre near Cambridge, UK. It is accessible at: `http://www.sanger.ac.uk/Software/Pfam/`

Contents. Pfam is a collection of protein domain sequences that are classified into families according to multiple sequence alignments and profile hidden Markov models. Hypertext links to external databases such as PDB (cf. A.75), PROSITE (cf. A.85), and InterPro (cf. A.50).

Data modelling, data storage, and data acquisition. The data are stored in several flat files (see [PFAMa] and Pfam web site for details). Most files are explicitly structured. There is e.g. one file containing all sequence data of the according proteins in FASTA-format [56], and one file containing the full alignments of the curated families.

Data in Pfam are acquired primarily from SWISS-PROT (cf. A.103). These data are processed to generate the Pfam database. [PFAMb] provides more information about the data acquisition and transformation.

Data retrieval and query answering. Pfam can be accessed via several complex fixed-form query interfaces. There are possibilities to query keywords or a protein sequence against the database. One interface allows the user to search for multiple organism names, several names can be combined using the boolean operators AND, OR, and NOT.

There is no *ad-hoc* query interface.

The classification of Pfam can also be browsed to retrieve single data items. In addition, the Pfam flat files can be downloaded via FTP at: `ftp://ftp.sanger.ac.uk/pub/databases/Pfam`

Size. All flat files are large and each require between 8 and 67 MB.

Literature on Pfam:

[PFAMa] A. Bateman et al. The Pfam Protein Families Database. *Nucleic Acids Research*, 28(1):263-266, 2000.

[PFAMb] `http://www.sanger.ac.uk/Software/Pfam/help/gstart.shtml`

A.79 PSD: PIR-International Protein Sequence Database (also known as PIR)

Provider and access. The PSD is offered by the Protein Information Resource (PIR) at the National Biomedical Research Foundation, USA, in collaboration with MIPS (cf. A.63) and the Japan International Protein Sequence Database (JIPID). The database is publicly accessible on the PIR web-site at:
`http://pir.georgetown.edu`
or at: `http://www.mips.biochem.mpg.de`.

Contents. The PSD is a resource for annotated protein sequences. Entries are classified into family groups and alignments of each group are available. Full releases are published quarterly (interim updates are publicly available weekly). In addition to PSD, PIR-International provides WWW access to other databases and maintains several internal data collections used for sequence annotation and integrity check:

- PATCHX: is a non-redundant database produced by MIPS of publicly available protein sequences that are not yet added to PSD. PSD and PATCHX is the most complete non-redundant collection of protein sequences available in the public domain.
- ARCHIVE: is a database of protein sequences as originally reported in a publication or submission.
- NRL_3D: is a sequence-structure database which is produced from sequence and annotation in the PDB (cf. A.75). It makes the sequences in the PDB available for similarity search and data retrieval.
- FAMBASE: is a collection of representative sequences from each protein family that can be used in a similarity search to reduce search time and improve sensitivity for identifying distant families.
- PIR-ALN: is a curated database of sequence alignments of super families, families and homology domains, with annotation information derived from PSD.
- RESID: is a database of post-translational modifications with descriptive, chemical, structural and bibliographic information.
- ProClass: is a protein family database that organises non-redundant PSD and SWISS-PROT (cf. A.103) sequences according to PIR super families and ProSite (cf. A.85) patterns. For detailed information about ProClass see A.83 at page 207.
- ProtFam: is a curated database of homology clusters with automatically generated multiple sequence alignments for families, super families and homology domains. See also MIPS (cf. A.63).

The PIR provides hot-links to other sequence databases, including MEDLINE (cf. A.59), PDB (cf. A.75), GDB (cf. A.35), OMIM (cf. A.72), FlyBase (cf. A.34), MIPS/Yeast (cf. A.63), SGD/Yeast (cf. A.100), MIPS/Arabidopsis (cf. A.63), and TIGR (cf. A.107) as well as context cross-references between its own database entries.

Data modelling, data storage, and data acquisition. The PIR databases are organised by an object-relational DBMS. A three-tier network computing architecture provides a framework for distributed object computing and Java-based WWW interfaces connect with the database server for queries and updates. There is no documentation found on the data model. The flat file release is separated in four data sets (annotated and classified entries only annotated entries, unverified entries, unencoded or untranslated entries).

New data are acquired by PIR from the literature, by translation of nucleic acid sequences submitted to GenBank (cf. A.36), EMBL (cf. A.28), and DDBJ (cf. A.22) or by submission to PIR-International. Submission can be done via an electronic fill-in-form, e-mail or PC-compatible disks. PIR accept submissed sequences only under few conditions (see PIR homepage). Annotations can be updated later in the same way like submission of new data.

Data retrieval and query answering. The data can be accessed directly after download via FTP at: `ftp://nbrfa.georgetown.edu/pr` or through search engines of three types:

- interactive text-based search engines which allow boolean queries of text fields. Output are links to all results. The results are HTML-representations of flat files structured in PSD-format.
- standard sequence similarity search engines, including Peptide Match, Pattern Match, BLAST [4], FASTA [56], Pairwise Alignment and Multiple Alignment. A graphical output interface is generated for each result with links to the HTML-representations of flat files structured in PSD-format.
- advanced search engines that combine sequence similarity with annotations searches or evaluate gene family relationships. A graphical output interface is generated as above.

Sequence searching is provided for PSD, NRL_3D, PATCHX, FAMBASE, ProClass and the combined PSD+PATCHX collections. Text and entry searching can be performed against PSD, NRL_3D, PIR-ALN, and RESID databases. Quarterly a CD-ROM called ATLAS is available. ATLAS includes the PSD, PIR-ALN, NRL_3D, RESID and PATCHX as well as database retrieval software.

There is no *ad-hoc* query interface nor a possibility to browse the data at PIR International.

Size. More than 150 000 sequences from many different organisms are available, more than 99% of the entries are classified by protein families and more than 57% are classified by protein super families.

Literature on PIR

Literature on PIR:

[PIRa] `http://www.mips.biochem.mpg.de/mips/more_cd_info.html`

[PIRb] W. Baker et al. The Protein Information Resource. *Nucleic Acids Research*, 28(1):41-44, 2000.

A.80 PLMItRNA: a Database for tRNAs and their genes in plant mitochondria

Provider and access. PLMItRNA is maintained by the University of Bari/Italy and freely accessible at: `http://bio-www.ba.cnr.it:8000/srs6/`

Contents. PLMItRNA is a database for tRNA molecules and genes identified in the mitochondria of all green plants. It reports information and multi-alignments on 254 genes and 16 tRNA molecules.

The current version of PLMItRNA database has been developed from a previous database for seed plant tRNA molecules and genes [PLMb]. It was updated in Summer 1999 by analysing the whole EMBL database using the Sequence Retrieval System (SRS, cf. Chapter 4.3.7).

Textual literature references are reported in the relative entries. There are no hypertext links to other databases.

Data modelling, data storage, and data acquisition. The data are stored in a flat file consisting of entries. Each entry is structured in 12 fields (line types). New data are acquired from EMBL (cf. A.28), the scientific literature, and submissions from the community.

Data retrieval and query answering. Data can be retrieved from PLMItRNA via a local SRS server. Several fixed-form query interfaces and a standard all text query form are provided.

The data can not be downloaded.

For similarity search the FASTA [56] program, which is with the database, can be used.

There is no *ad-hoc* query interface nor a possibility to browse the data.

Size. PLMItRNA contains 452 entries for 436 genes and 16 tRNA sequences.

Literature on PLMItRNA:

[PLMa] V. Volpetti et al. PLMItRNA, a Database for tRNAs and tRNA Genes in Plant Mitochondria: Enlargements and Updating. *Nucleic Acids Research*, 28(1):77-80, 2000.

[PLMb] L. Ceci et al. PLMItRNA, a Database for tRNAs and tRNA Genes in Plant Mitochondria. *Nucleic Acids Research*, 27(1):156-157, 2000.

A.81 PombePD: The Pombe Proteome Database

Provider and access. The Pombe Proteome Database (PombePD) is maintained by Proteome Inc. at Beverly/USA and accessible freely to academic users at: `http://www.proteome.com/databases/index.html`
Proteome Inc. also provides WormPD (cf. A.116) and YPD (cf. A.119).

Contents. PombePD stores proteome information about the proteins of the pombe model organism. Hypertext links to several other databases such as, among others, GenBank (cf. A.36), SWISS-PROT (cf. A.103), MIPS (cf. A.63), and PubMed (cf. A.90) are provided.

Data modelling, data storage, and data acquisition. PombePD is based on a relational model and uses the commercial software Oracle as DBMS. Further information about the data model is not provided.

A controlled vocabulary for querying and analysing data in PombePD is enforced.

Data are gathered from the scientific literature and submissions of the community.

Data retrieval and query answering. The data are accessible via a (complex) fixed-form query interfaces or an advanced full-text keyword search (several keywords can be combined by using the boolean operators "AND"/"OR"). PombePD can also be searched by the gene name encoding a protein. The database can also be browsed starting at [PomDb], following links classified by several categories to at least a list of all data items in the according category.

The data items are represented in a special format called the "Protein Report". The Protein Report lists all information about a protein in a HTML table. A comprehensive reference list with links to their MEDLINE (cf. A.59) abstracts are provided.

There is no *ad-hoc* query interface nor a flat file release available.

Size. The current release (November 2000) stores information about 4787 proteins and provides 1888 references. The size of the data in MB is not available.

Literature on PombePD:

[PomDa] M. Costanzo et al. The Yeast Proteome Database and Caenorhabditis elegans Proteome Database: comprehensive resources for organisation and comparison of model organism protein information. *Nucleic Acids Research*, 28(1):73-76, 2000.

[PomDb] `http://www.proteome.com/databases/index.html`

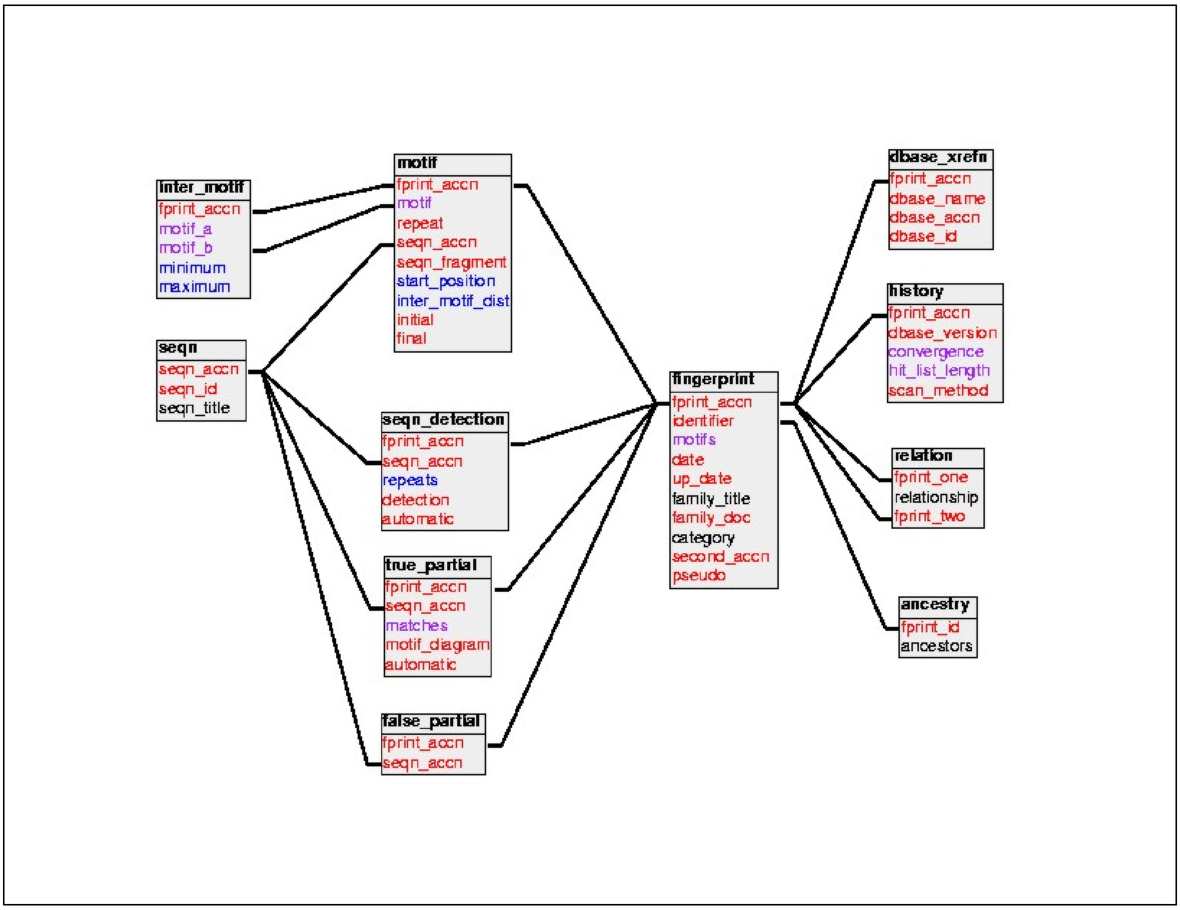


FIGURE A.24: Data model of the PRINTS-S database (available at: `http://bioinf.man.ac.uk/dbbrowser/sprint/printss_data_model.html`)

A.82 PRINTS-S formerly known as PRINTS

Provider and access. PRINTS-S is maintained at the University of Manchester, UK. It can be accessed at: `http://www.bioinf.man.ac.uk/dbbrowser/PRINTS/`

Contents. PRINTS-S is a collection of protein family fingerprints. These are groups of motifs that together are diagnostically more potent than single motifs.

Hypertext links to PubMed (cf. A.90) and SWISS-PROT (cf. A.103) are provided.

Data modelling, data storage, and data acquisition. PRINTS the predecessor of PRINTS-S was implemented as a single flat file. The new version called PRINTS-S is now rebuilt with an object-relational schema. The data model is shown in Figure A.24.

Data are acquired from other databases, primarily SWISS-PROT (cf. A.103).

Data retrieval and query answering. PRINTS-S allows all four common ways of data retrieval in molecular biology databases:

Several complex fixed-form query interfaces are available as well as an *ad-hoc* query interface (query language seemed to be home-made but is documented).

PRINTS-S provides a classification of the data items to be browsed for data retrieval.

In addition, the entire database is available as a file (in object-relational format). The sequences stored in PRINTS-S are available in a separate file (FASTA [56] format). These files can be downloaded via FTP at: `ftp://bioinf.man.ac.uk/pub/prints/`

Size. The data file of PRINTS-S requires approximately 31 MB.

Literature on PRINTS-S:

[PRINa] T. Attwood et al. PRINTS-S: The Database Formerly Known as PRINTS. *Nucleic Acids Research*, 28(1):225-227, 2000.

[PRINb] `http://bioinf.man.ac.uk/dbbrowser/prints`

A.83 ProClass

Provider and access. ProClass is maintained by the Protein Information Resource (PIR) in Washington, USA. PIR provides several other databases (cf. A.79). ProClass can be accessed at: `http://pir.georgetown.edu/gfserver/proclass.html`

Contents. ProClass is a protein family database that organises non-redundant sequence entries into families defined collectively by PIR (cf. A.79) super-families and PROSITE (A.85) patterns.

Data modelling, data storage, and data acquisition. Data are stored in an object-relational version of the commercial DBMS Oracle. There is no information about the underlying schema available. In addition, the data are stored in several flat files for distribution. A list of these files can be obtained from [PROCLa], their formats are described in full detail at the ProClass homepage.

New data are derived from other databases, primarily PIR (cf. A.79), PROSITE (cf. A.85), and SWISS-PROT (cf. A.103).

Data retrieval and query answering. A fixed-form query interface provides keyword search against special database objects. The database can also be searched by sequence similarity and family classification. A flat file release as mentioned above can be downloaded via FTP.

There is no *ad-hoc* query interface. ProClass data cannot be browsed.

Size. ProClass contains more than 155 000 sequence entries. The zipped data files all together require about 55 MB.

Literature on ProClass:

[PCLa] H. Huang, C. Xiao, and C. Wu. ProClass Protein Family Database. *Nucleic Acids Research*, 28(1):273-276, 2000.

[PCLb] `http://pir.georgetown.edu/gfserver/pc_intro.html`

A.84 ProDom and ProDom-CG

Provider and access. ProDom and ProDom-CG are maintained at INRA/CNRS in France.
ProDom can be accessed at: `http://www.toulouse.inra.fr/prodom.html`
ProDom-CG can be accessed at: `http://www.toulouse.inra.fr/prodomCG.html`

Contents. ProDom contains all protein domain families automatically generated from SWISS-PROT (A.103). ProDom-CG results from a similar domain analysis as applied to completed genomes. Hypertext links to other databases are provided.

Data modelling, data storage and data acquisition. There is no information about the data model and it's implementation. In all likelihood, the data are stored in flat files. Data are acquired automatically from SWISS-PROT and TrEMBL (cf. A.103). For detailed information about the building of ProDom see [PRODa].

Data retrieval and query answering. ProDom and ProDom-CG provide fixed-form query interfaces. Both databases can be searched for sequence similarity using BLAST [4]. There are no *ad-hoc* query interfaces. The data cannot be browsed.

Data can also be retrieved as a flat file by download via FTP or via the Sequence Retrieval System (SRS, cf. Chapter 4.3.7).

Size. The (zipped) flat files require between 16 MB (BLAST format) and 51 MB (SRS format).

Literature on ProDom/ProDom-C:

[PRODa] F. Corpet et al. ProDom and ProDom-CG: Tools for Protein Domain Analysis and Whole Genome Comparisons. *Nucleic Acids Research*, 28(1):267-269, 2000.

[PRODb] `http://www.toulouse.inra.fr/prodom/doc/www_info.html`

A.85 PROSITE

Provider and access. The PROSITE database is developed and maintained at the Swiss Institute for Bioinformatics at Geneva. It can be accessed at: `http://www.expasy.ch/prosite/`

Contents. PROSITE consists of biologically significant protein profiles and patterns. New protein sequences can be classified to an existing family (if matching) or building a new class of proteins. Hypertext links to SWISS-PROT (cf. A.103) are provided.

Data modelling, data storage, and data acquisition. PROSITE is a flat file database consisting of two files. The first file (named PROSITE.DAT) consists of entries that are explicitly structured with search indexes (see [PROSa] for a sample) and is not well human readable but in return it contains all information necessary for computer programs that makes use of PROSITE.

The second file (named PROSITE.DOC) contains textual information that documents each pattern. Entries in that file are mostly unstructured. Each entry is rather structured like an article.

Data seem to be acquired from SWISS-PROT (cf. A.103). Profiles for new SWISS-PROT sequences are generated using sequence alignment as basis for classification.

Data retrieval and query answering. The PROSITE database provides several fixed form query interfaces for data retrieval. Some search indexes of the PROSITE.DAT file can be searched for keyword. A full-text keyword search is also possible. There is no *ad-hoc* query interface, but PROSITE can be accessed via the Sequence Retrieval System (SRS cf. Chapter 4.3.7).

In addition, the entire flat files (PROSITE.DAT and PROSITE.DOC) can be downloaded via FTP at: `ftp://www.expasy.ch/ftp/databases/prosite`

Size. The PROSITE.DAT file requires approximately 4.0 MB and the PROSITE.DOC file requires approximately 2.4 MB.

Literature on PROSITE:

[PROSa] K. Hofmann et al. The PROSITE database, its status in 1999. *Nucleic Acids Research*, 27(1):215-219, 1999.

[PROSb] `http://www.expasy.ch/prosite/`

A.86 ProtFam

see MIPS (A.63 at page 186) or PIR (A.79 at page 202)

A.87 ProTherm:The Thermodynamic Database for Proteins and Mutants

Provider and access. The Thermodynamic Database for Proteins and Mutants (ProTherm) is maintained at the Institute of Physical and Chemical Research (RIKEN), Japan. It is accessible at: `http://www.rtc.riken.go.jp/jouhou/Protherm/protherm.html`

Contents. ProTherm provides numerical data for several thermodynamic parameters, structural information, experimental methods and conditions, functional and literature information. There are hypertext links to several other structure and literature databases.

Data modelling, data storage, and data acquisition. There is no information about the data model and implementation of ProTherm available. We guess that it is a flat file database. There is no information about the data source.

Data retrieval and query answering. ProTherm provides a complex fixed-form query interface. Data can only be retrieved via this query form. There is no *ad-hoc* interface, nor a possibility to browse the data, nor flat file release for download.

Size. ProTherm currently consists of 7114 entries.

Literature on ProTherm:

[PROTHa] M. Gromiha et al. ProTherm Version 2.0: Thermodynamic Database for Proteins and Mutants. *Nucleic Acids Research*, 28(1):283-285, 2000.

[PROTHb] `http://www.rtc.riken.go.jp/jouhou/Protherm/protherm.html`

A.88 ProtoMap

Provider and access. ProtoMap is maintained at the Hebrew University at Jerusalem/Israel. It can be accessed at: `http://www.protomap.cs.huji.ac.il/search.html`

Contents. The ProtoMap site offers an exhaustive classification of all proteins in the SWISS-PROT (cf. A.103) database.

Data modelling, data storage, and data acquisition. There is no information about the model and implementation of the database. It seems that the proteins are classified, and each cluster stores links to the SWISS-PROT (cf. A.103) entries of its members. The protein space (all known proteins in

SWISS-PROT) is represented as a weighted graph whose vertices are the sequences. The weight of an edge between two sequences corresponds to their degree of similarity.

Data are acquired from the SWISS-PROT (cf. A.103) database. The classification procedure is described in full detail in [PMAPb]. ProtoMap updates are associated with major SWISS-PROT updates.

Data retrieval and query answering. ProtoMap can be searched via a fixed-form query interface. Data can be extracted by searching for keyword, SWISS-PROT ID/AC, protein name, cluster number, or cluster size. Clusters of proteins can also be browsed to retrieve data items.

There is no *ad-hoc* interface nor a flat file release.

Size. No information available.

Literature on ProtoMap:

[PMAPa] G. Yona, N. Linial, and M. Linial. ProtoMap: Automatic Classification of Protein Sequences and Hierarchy of Protein Familie. *Nucleic Acids Research*, 28(1):49-55, 2000.

[PMAPb] G. Yona et al. A Map of the Protein Space : An Automated Hierarchical Classification of all Protein Sequences. In *Proccedings of the 6th International Conference on Intelligent Systems in Molecular Biology (ISMB)*, 1998.

[PMAPc] `http://www.protomap.cs.huji.ac.il/search.html`

A.89 PseudoBase

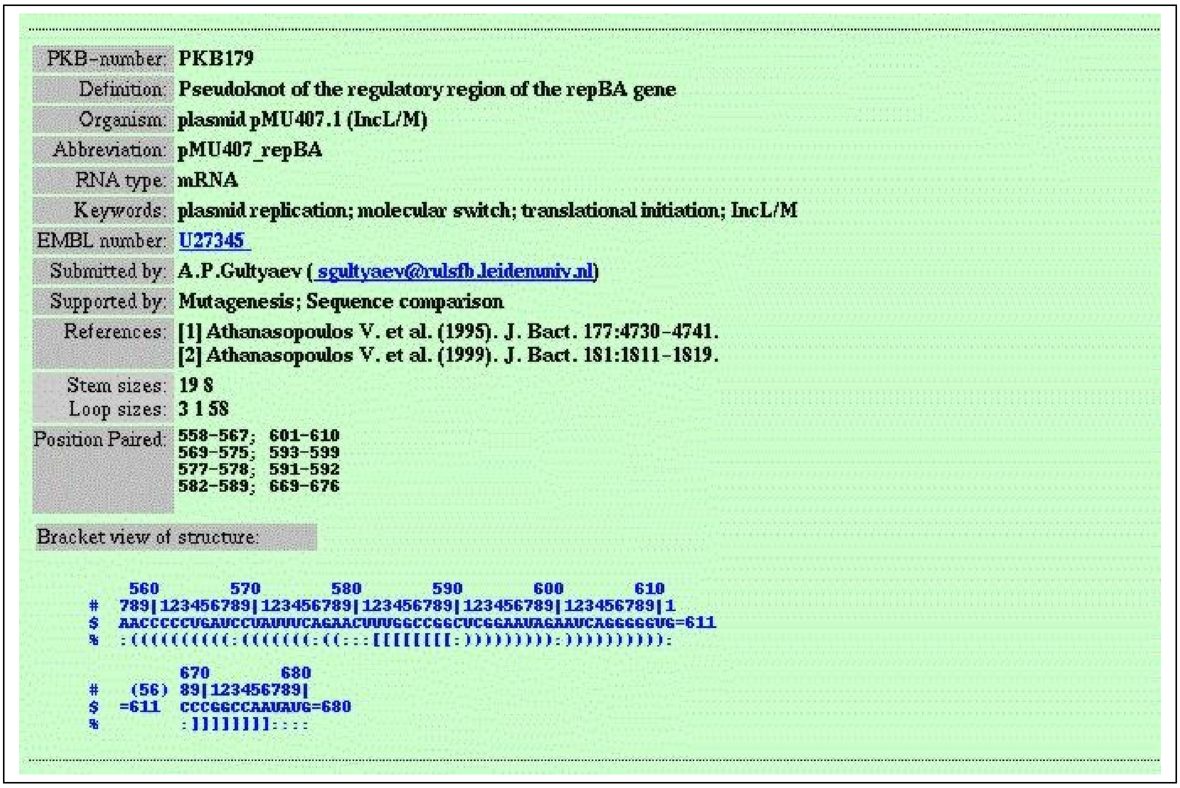
Provider and access. PseudoBase is maintained by the Leiden University at Leiden/The Netherlands. It can be accessed at: `http://wwwbio.LeidenUniv.nl/~Batenburg/PKB.html`.

Contents. PseudoBase is a database containing structural, functional, and sequence data related to RNA pseudoknots. PseudoBase provides hypertext links to EMBL (cf. A.28).

Data modelling, data storage, and data acquisition. The data are stored in one or more flat files. Each entry contains 12 items, listed in [PseuBa] can be derived from Figure A.89.

Data are acquired from the research community. There is a web-based form for submission of new data with the database.

Data retrieval and query answering. Data can be retrieved by browsing stepwise through a hierarchical tree. There are no query interfaces and no flat files for download.



PKB-number:	PKB179
Definition:	Pseudoknot of the regulatory region of the repBA gene
Organism:	plasmid pMU407.1 (IncL/M)
Abbreviation:	pMU407_repBA
RNA type:	mRNA
Keywords:	plasmid replication; molecular switch; translational initiation; IncL/M
EMBL number:	U27345
Submitted by:	A.P.Gultyaev (sgultyaev@rulsfb.leidenuniv.nl)
Supported by:	Mutagenesis; Sequence comparison
References:	[1] Athanasopoulos V. et al. (1995). J. Bact. 177:4730-4741. [2] Athanasopoulos V. et al. (1999). J. Bact. 181:1811-1819.
Stem sizes:	19 8
Loop sizes:	3 1 58
Position Paired:	558-567; 601-610 569-575; 593-599 577-578; 591-592 582-589; 669-676

Bracket view of structure:

```
      560       570       580       590       600       610
#  789|123456789|123456789|123456789|123456789|123456789|1
$  AACCCCCUGAUCCUAUUUCAGAACUUUGGCCGGCUCGGAAUAGAAUCAGGGGGUG=611
%  :((((((((((:(((((((:((:::[[[[[[[[:)))))))))):)))))))))))):

            670       680
#   (56)  89|123456789|
$  =611   CCCGGCCAAUAUG=680
%         :]]]]]]]]::::
```

FIGURE A.25: Representation of a sample entry in PseudoBase. The 12 different data items recorded in each record are shown.

Size. No information available.

Literature on PseudoBase:

[PseuBa] F. Batenburg et al. PseudoBase: a Database With RNA Pseudoknot. nar , 28(1):201-204, 2000.

[PseuBb] `http://wwwbio.LeidenUniv.nl/˜Batenburg/PKBAbout.html`

A.90 PubMed

Provider and access. PubMed is maintained at the National Center for Biotechnology Information (NCBI) at Bethesda, USA. It can be accessed at: `http://www.ncbi.nlm.nih.gov/PubMed/`

Contents. PubMed was developed in conjunction with publishers of biomedical literature as a search tool for accessing literature citations and linking to full-text journals at web sites of participating publishers. The former service of NCBI called MEDLINE (cf. A.59) is re-implemented in PubMed. There are no links to other molecular biology databases.

Data modelling, data storage, and data acquisition. There is no information available about the implementation of PubMed. Most likely the data are modelled using ANS.1 [5] the exchange format of NCBI. Thus, PubMed is probably a flat file database. An ontology called MeSH is used for indexing articles in PubMed.

There is no information available where data in PubMed are acquired.

Data retrieval and query answering. PubMed is accessible via Entrez (cf. Chapter 4.3.4), the integrated retrieval system of NCBI. It provides several fixed-form query interfaces (keyword search against special search indexes). There is no *ad-hoc* query interface nor a flat file release provided for download.

In addition, there is a tool called "Journal Browser" that supports browsing through the list of available journals within PubMed.

Size. No information available.

Literature on PubMed:

[PMa] `http://www.ncbi.nlm.nih.gov/PubMed/`

A.91 RDP: The Ribosomal Database Project

Provider and access. The Ribosomal Database Project is hosted by the Michigan State University. A database is offered at: `http://www.cme.msu.edu/RDP`

Contents. The database of RDP offers ribosomal RNA sequence data associated with analysis tools. There is no information about interconnections with other databases available.

Data modelling, data storage, and data acquisition. Data is stored and managed by the commercial object DBMS ObjectStore. Information about the object data model is not available.

Data are acquired from GenBank (cf. A.36) and EMBL (cf. A.28) and from direct submissions to RDP.

Data retrieval and query answering. Data can only be accessed as flat files: A large amount of different files can be downloaded via FTP, including data files with "small subunit rRNA sequences", "large subunit rRNA sequences", user submitted alignments, and several source files for analysis tools. Data files in GenBank (cf. A.36) format are also available.

There are no query interfaces at all nor any possibilities to browse the data.

Several analysis tools (primarily tools for sequence similarity searches) are with the database (see [RDPa] and [RDPb]).

Size. The files differ in size between around 100 KB up to more than 20 MB. There is no information about the size of the entire database.

Literature on RDP:

[RDPa] B. Maidak et al. The RDP (Ribosomal Database Project) Continues. *Nucleic Acids Research*, 28(1):173-174, 2000.

[RDPb] `http://www.cme.msu.edu/RDP/docs/documentation.html`

[RDPc] B. Maidak et al. A new Version of the RDP (Ribosomal Database Project). *Nucleic Acids Research*, 27(1):171-173, 1999.

A.92 REBASE: The Restriction Enzymes Database

Provider and access. The Restriction Enzymes Database (REBASE) is maintained at the New England BioLabs, Inc in Beverly, USA. It can be accessed at: `http://rebase.neb.com/rebase/rebase.html`

Contents. REBASE is a comprehensive database of information about restriction enzymes and related proteins. It provides hypertext links to GenBank (cf. A.36).

Data modelling, data storage, and data acquisition. There is no information available about the underlying data model and the data management system. We suggest that it is implemented as a relational DBMS. In fact, REBASE is available as flat file database in several formats (see [REBAb] for comprehensive list of available flat file formats).

Data seem to be acquired primarily from GenBank (cf. A.36). A submission tool enables researchers to add new data to REBASE.

Data retrieval and query answering. REBASE provides fixed form query interfaces where keywords can be queried against several search categories. There is no *ad-hoc* query interface. The data in REBASE are also structured into a hierarchy that can be browsed for data retrieval.

In addition, the REBASE database can be downloaded as flat file via FTP. The database is provided in several formats (see [REBAb]) fitting as input for several different analysis tools.

Note: Some data items or some parts of single data items are only commercially available.

Size. No information available.

Literature on REBASE:

[REBAa] R. Roberts and D. Macelis. REBASE – Restriction Enzymes and Methylases. *Nucleic Acids Research*, 28(1):306-307, 2000.

[REBAb] `http://rebase.neb.com/rebase/rebase.files.html`

A.93 RegulonDB

Provider and access. RegulonDB is maintained at UNAM in Morelos/Mexico in a collaboration with the University of Wisconsin in Madison/USA. It is freely accessible at: `http://www.cifn.unam.mx/ComputationalBiology/regulondb/`

Contents. RegulonDB is a database on transcription regulation and operon organisation in *E. coli* Protein sequences are hypertext linked to SWISS-PROT (cf. A.103) and all entries are hypertext linked to bibliographic references via MEDLINE (cf. A.59).

Data modelling, data storage, and data acquisition. The data are built upon a relational model. Originally, RegulonDB was developed using a commercial software called 4^{th}Dimension (4D) in Macintosh platform. Figure A.26 presents this original schema of RegulonDB. [REDBb] contains more detailed information about the original schema which was slightly modified since (see [REDBa] for information on this modifications and on the up-to-date schema).

Currently RegulonDB is implemented using the commercial relational DBMS Oracle 8. Forms to access data from the web have been implemented using PL-SQL query language of Oracle.

Data are acquired from the scientific literature and several other databases.

Data retrieval and query answering. Data can be accessed via a fixed-form query interface in several different ways:

- Search by object: the database can be searched against several explicit objects such as gene, regulon, protein, etc. by name and accession number.
- Search by region: the database can be searched for objects which are within a special region of the genome

The data are represented in HTML tables or as interactive maps (search by region).

There are no *ad-hoc* query interfaces, nor any capabilities to browse the data, nor a flat file release for downloading.

Size. The database currently stores over 20 000 objects. See [REDBc] for detailed information (e.g. the current number of instances of each table.

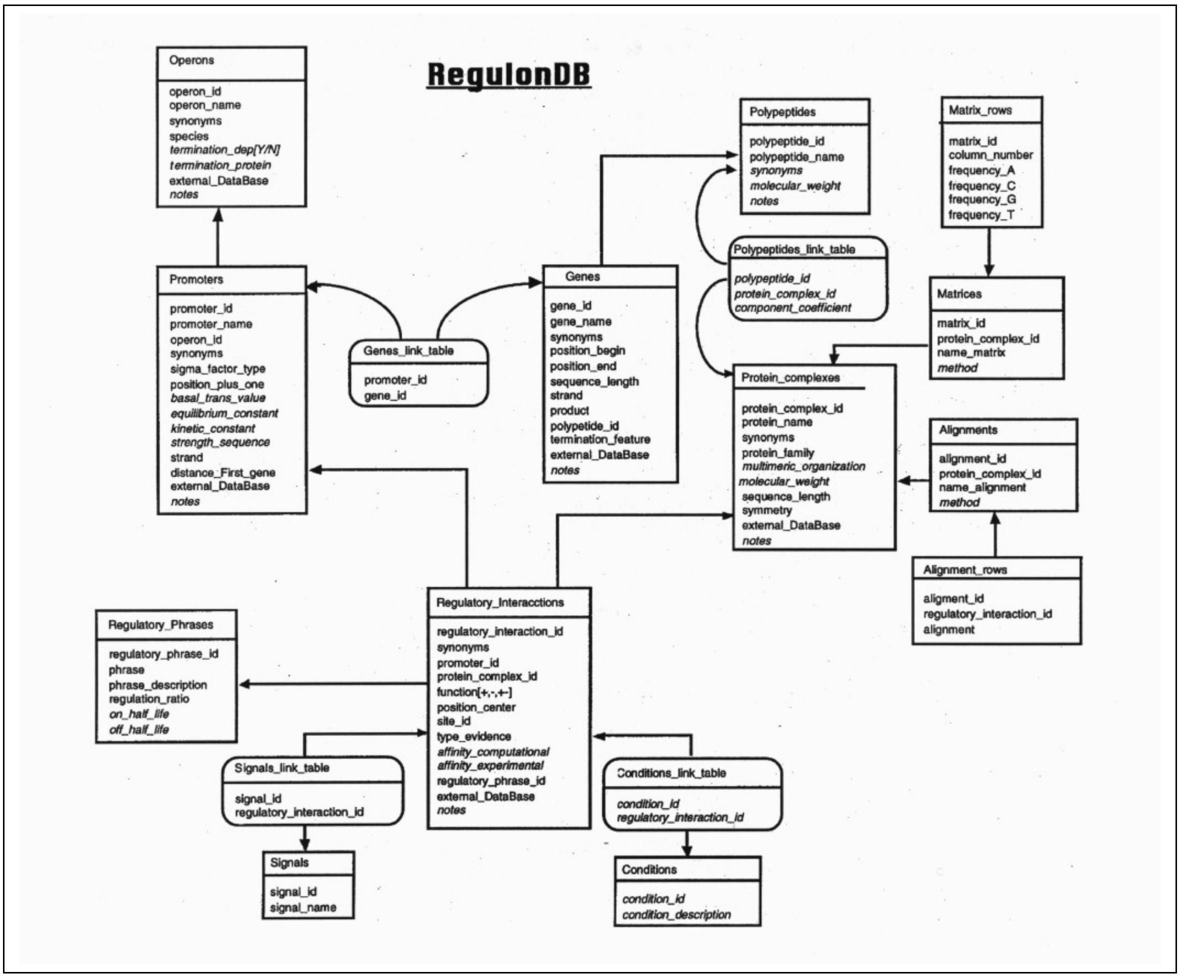


FIGURE A.26: Original schema of RegulonDB (taken from [REDBb]).

Literature on RegulonDB:

[REDBa] H. Salgado et al. RegulonDB: Transcriptional Regulation and Operon Organisation in E. coli K-12. *Nucleic Acids Research*, 28(1):65-67, 2000.

[REDBb] A. Huerta et al. RegulonDB: Transcriptional Regulation and Operon Organisation in E. coli K-12. *Nucleic Acids Research*, 26(1):55-59, 1998.

[REDBc] `http://www.cifn.unam.mx/Computational_Biology/regulondb/docs`

A.94 RHdb: The Radiation Hybrid Database

Provider and access. The Radiation Hybrid Database (RHdb) is maintained by the the European Bioinformatics Institute (EBI) at Cambridge/UK. It is accessible at: http://www.ebi.ac.uk/RHdb

Contents. RHdb stores radiation hybrid data. It is cross referenced via hypertext links with several other databases (a current, comprehensive list can be obtained at the RHdb homepage).

Data modelling, data storage, and data acquisition. The schema of RHdb is based on an object model. This model is then manually mapped onto the underlying relational DBMS Oracle. A tool called "Persistence"[5] is used to generate C++ classes from the relational schema. "Persistence" generates C++ classes and methods reflecting the tables and relationships in the relational schema, as well as standard methods to access the relational tables: INSERT, DELETE, SELECT, UPDATE. With the aid of "Persistence", only the more domain specific queries need to be implemented by hand, the rest is generated automatically. The reason for this complex architecture is that this supports the development of a CORBA [31] server on top of the relational DBMS (the version of Oracle used for RHdb does not provide this automatically). In Chapter 4.3.3 we described the approach of EBI to integrate heterogeneous databases using CORBA.

The object oriented schema can be browsed at [RHDBb].

Data are acquired from the research community (submissions should follow a specific format which is described at the RHdb homepage).

Data retrieval and query answering. RHdb data can be retrieved through several fixed-form query interfaces. In fact these search tools are CORBA [31] clients using Java applets. In addition the database can be searched using the Sequence Retrieval System (SRS, cf. 4.3.7). A SRS server is accessible at EBI.

RHdb also produces a flat file release which can be downloaded via FTP from EBI. The release contains four files with different formats (described at the RHdb homepage).

There is no *ad-hoc* query interface nor any possibility to browse the data.

Size. All together the four files require currently about 94 MB (zipped about 11 MB).

Literature on RHdb:

[RHDBa] P. Rodriguez-Tome and P. Lijnzaad. RHdb: The Radiation Hybrid Database. *Nucleic Acids Research*, 28(1):146-147, 2000.

[RHDBb] http://corba.ebi.ac.uk/RHdb/schema/RHdb_object.html

[5] http://www.persistence.com

A.95 SacchDB: The macintosh Version of the *S. cerevisiae* Genome Database (SGD)

Provider and access. SacchDB is the Macintosh version of the *Saccharomyces* Genome Database (SGD, see page 221). It can be downloaded via FTP for local use at: `genome-ftp.stanford.edu`
and at: `ncbi.nlm.nih.gov`

A UNIX version is now also available. SacchDB is maintained by the Stanford University at Stanford/USA.

Contents. The *Saccharomyces* Genome Database (SGD, cf. A.100) stores and organises information about all of the nearly 6200 genes in the yeast genome.

SGD focuses on providing comprehensive reviews of the literature of each gene.

Data modelling, data storage, and data acquisition. SacchDB is implemented using the ACEDB software as DBMS. The schema can be browsed using the query interface. Data are acquired from the scientific community and the literature.

Data retrieval and query answering. The common ways of querying an ACEDB-database are provided: Fixed-form queries, *ad-hoc* queries and simple text search can be performed. Data can also be retrieved as flat files.

In addition, the database schema can be browsed to retrieve single data items.

Size. No information available.

Literature on SacchDB:

[Sacch] `http://genome-www.stanford.edu/Saccharomyces/aboutSacchDB.html`

A.96 SBASE

Provider and access. SBASE is maintained and provided by the International Centre for Genetic Engineering and Biotechnology at Trieste/Italy. It is accessible at: `http://www.icgeb.trieste.it/sbase/`
and `http://sbase.abc.hu/sbase`

Contents. SBASE contains annotated segments of protein sequences. It is cross-referenced through hypertext links to all major sequence databases such as GenBank (cf. A.36), EMBL (cf. A.28), SWISS-PROT (cf. A.103), etc., and other specialised databases (see [SBAa] for a comprehensive list).

Data modelling, data storage, and data acquisition. Data are kept in flat files. The structure of the data entries follows that of the EMBL (cf. A.28) and SWISS-PROT (cf. A.103) databases.

Data originate from three main sources: SWISS-PROT (cf. A.103), PSD/PIR (cf. A.79), and the scientific literature. Currently new data are acquired also from GenBank (cf. A.36) and EMBL (cf. A.28) by translating the nucleotide sequences into amino acid sequences.

Data retrieval and query answering. SBASE can be searched by BLAST [4] via the web interfaces. The data can also be downloaded via FTP [SBAa].

There are no other fixed-form or *ad-hoc* query interfaces nor any capabilities to browse the data.

Size. As of Winter 1999, the complete database is 221 MB and its compressed form is 16.3 MB.

Literature on SBASE:

[SBAa] J. Murvai et al. The SBASE Protein Domain Library, Release 7.0: a Collection of Annotated Protein Sequence Segments. *Nucleic Acids Research*, 28(1):260-262, 2000.

[SBAb] `http://www.icgeb.trieste.it/sbase/`

A.97 SCOP: A Structural Classification of Proteins Database

Provider and access. The Structural Classification of Proteins database (SCOP) is provided by the MRC laboratory of Molecular Biology at Cambridge/UK. It can be accessed at: `http://scop.mrc.lmb.cam.ac.uk/scop/`

Contents. SCOP stores a detailed and comprehensive description of the relations of known protein structures. The classification of the proteins is described in more detail in [SCOPa]. SCOP provides hypertext links to several other databases (see [SCOPb]).

Data modelling, data storage, and data acquisition. SCOP is available as a set of tightly coupled hypertext pages. There is no information provided about the data model and the data management system. But it seems that the data are stored internally in flat files. Data are acquired from the PDB (cf. A.75)

Data retrieval and query answering. The classification hierarchy of SCOP can be browsed. A simple keyword search form is also provided. There is no *ad-hoc* or complex fix-form query interface. Several flat files containig the SCOP data can be downloaded via FTP.

Size. Currently SCOP provides 11410 PDB Entries, 26219 Domains. and 40 Literature References.

Literature on SCOP:

[SCOPa] L. Lo Conte et al. SCOP: a Structural Classification of Proteins database. *Nucleic Acids Research*, 28(1):257-259, 2000.

[SCOPb] `http://scop.mrc-lmb.cam.ac.uk/scop/help.html`

A.98 SELEXdb

Provider and access. SELEXdb is maintained at the Siberian Branch of the Russian Academy of Science at Novosibirsk, Russia. It can be accessed at: `http://wwwmgs.bionet.nsc.ru/mgs/systems/selex/`

Contents. SELEXdb contains DNA/RNA sequences and accumulated experimental data on functional sites. It provides no links to external databases.

Data modelling, data storage, and data acquisition. SELEXdb is a flat file database. The entries are explicitly structured using line types (search indexes). The structure of an entry is documented e.g. in [SELa].

Data are acquired from the scientific literature.

Data retrieval and query answering. SELEXdb is only accessible via a local Sequence Retrieval System (SRS, cf. 4.3.7) server. There is no local web interface and no possibility to download the data via FTP as flat files.

Size. No information available.

Literature on SELEXdb:

[SELa] J. Ponomarenko. SELEXdb: An Activated Database on Selected Randomized DNA/RNA Sequences Addressed to Genomic Sequence Annotation. *Nucleic Acids Research*, 28(1):205-208, 2000.

[SELb] `http://wwwmgs.bionet.nsc.ru/mgs/systems/selex/`

A.99 SENTRA

Provider and access. SENTRA is provided by the Argonne National Laboratory in Argonne, USA. It is accessible at: `http://wit.mcs.anl.gov/WIT2/Sentra/HTML/sentra.html`

Contents. SENTRA is a database of proteins associated with microbial signal transduction. Each entry is directly linked to the corresponding entry in SWISS-PROT/TrEMBL (cf. A.103) and WIT (cf. A.115).

Data modelling, data storage, and data acquisition. SENTRA is most likely a flat file database. There is no information available about the structure of the entries but they are most likely structured explicitly through line types.

Data are acquired from external databases.

Data retrieval and query answering. SENTRA can be accessed via a mixed type interface combining a fixed-form query interface with a browsable hierarchy.

There is no *ad-hoc* query interface nor a possibility to download the data as flat file via FTP.

Size. No information available.

Literature on SENTRA:

[SENa] M. D'Souza, M. Romine, and N. Maltsev. SENTRA: A Database of Signal Transduction Proteins. *Nucleic Acids Research*, 28(1):335-336, 2000.

A.100 SGD: *Saccharomyces* Genome Database

Provider and access. The *Saccharomyces* Genome Database (SGD) is maintained by the Stanford University at Stanford/USA and can be accessed freely at: `http://genome-www.stanford.edu/Saccharomyces/`.

Contents. SGD stores and organises information about all of the nearly 6200 genes in the yeast genome. SGD focuses on providing comprehensive reviews of the literature of each gene. A Macintosh version of SGD for local use (called SacchDB, cf. A.95) is available for download via FTP at:
`genome-ftp.stanford.edu`
and at: `ncbi.nlm.nih.gov`

Several hypertext links to other databases such as YPD (cf. A.119), MIPS (cf. A.63), SWISS-PROT (cf. A.103), and PSD/PIR (cf. A.79) are provided.

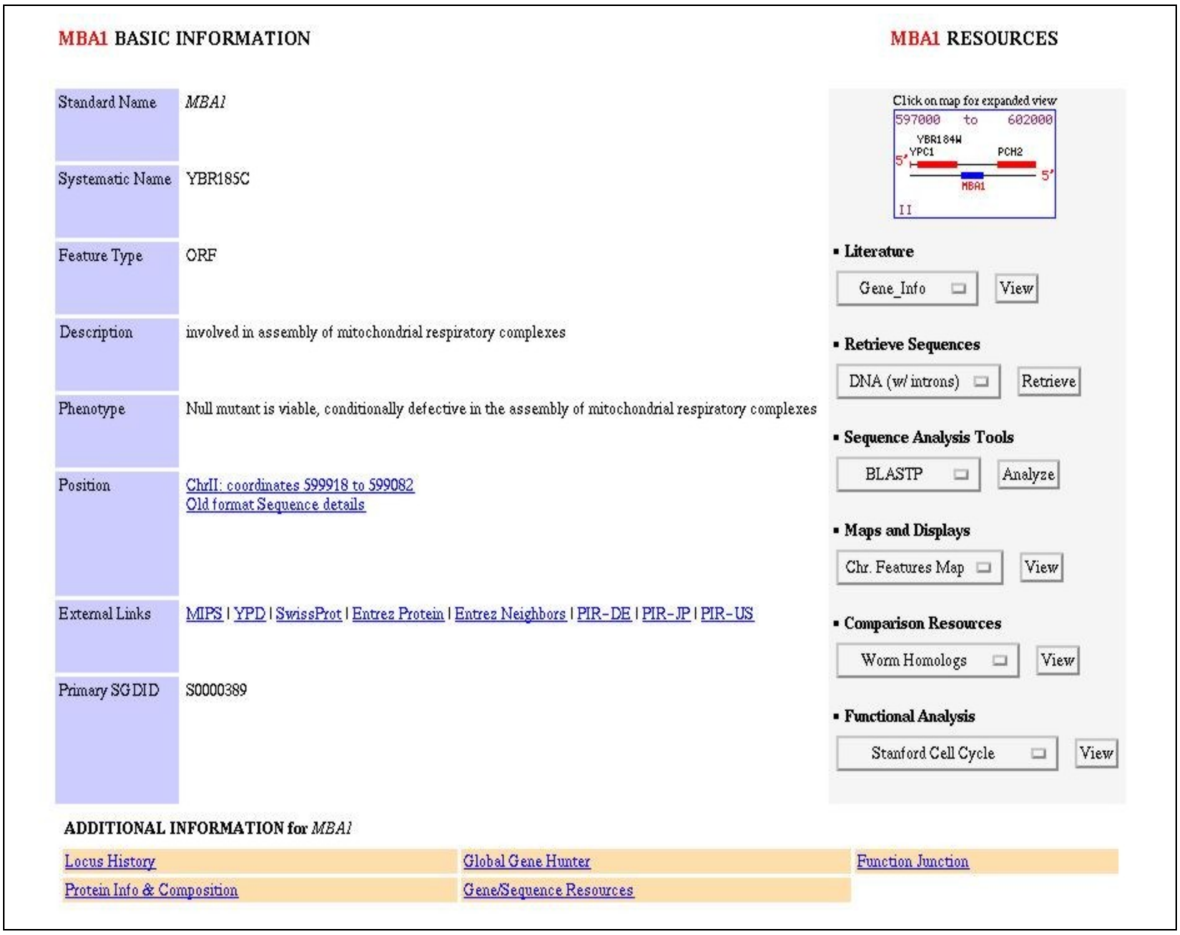


FIGURE A.27: Locus Page of a sample entry in SGD.

Data modelling, data storage, and data acquisition. SGD is currently re-implemented using the relational DBMS Oracle. The data model specification (table specification) can be browsed at [SGDb].

In collaboration with FlyBase (cf. A.34) and MGD (cf. A.62), SGD is creating the so called "Gene Ontology". This ontology will describe the important components and relationships that reflect the SGD's current understanding of cell biology.

Data are acquired from the scientific community and the literature.

Data retrieval and query answering. SGD provides fixed-form query interfaces to retrieve data. The information about each gene in the database is organised by the so called ”locus page”. Users will be able to retrieve a large set of data about any gene with minimal navigation, starting from the according locus page. The information is represented “in a format intuitive to biologists” ([SGDa]). Several fixed-form query interfaces and standard analysis tools are provided for further analysis and data retrieval. Figure A.27 shows a sample locus page.

The data on the locus page are also available in tab-delimited and XML formats to facilitate automated data exchange.

Ad-hoc queries can be performed using the ACEDB query language against the ACEDB version of SGD, SacchDB (cf. A.95).

SGD provides a flat file release for download via FTP. The ACEDB version SacchDB can be browsed to retrieve single data items.

Size. No information available.

Literature on SGD:

[SGDa] C. Ball et al. Integrating Functional Genomic Information into the Saccharomyces Genome Database. *Nucleic Acids Research*, 28(1):77-80, 2000.

[SGDb] `http://genome-www.stanford.edu/Saccharomyces/sgd_docs.html`

A.101 SMART: Simple Modular Architecture Research Tool

Provider and access. The SMART database (SMART) is maintained at the European Molecular Biology Laboratory (EMBL) in Heidelberg, Germany in collaboration with the University of Oxford, UK. It can be accessed at: `http://smart.embl-heidelberg.de/`

Contents. SMART allows the identification and annotation of genetically mobile domains and the analysis of domain architectures. The domains are extensively annotated with respect to phyletic distributions, functional class, tertiary structures and functionally important residues.

Hypertext links to several external databases (e.g. PROSITE (cf. A.85) and Pfam (cf. A.78) are provided.

Data modelling, data storage, and data acquisition. SMART is implemented using a relational DBMS. The underlying database schema is documented in [SMARa].

Data are acquired from other databases.

Data retrieval and query answering. SMART can be accessed through several fixed-form query interfaces. There is neither a *ad-hoc* query interface, nor a flat file release for download, nor a browsable classification of the data.

Size. No information available.

Literature on SMART:

[SMARa] J. Schultz et al. SMART: a Web-Based Tool for the Study of Genetically Mobile Domains. *Nucleic Acids Research*, 28(1):231-234, 2000.

[SMARb] `http://smart.embl-heidelberg.de/`

A.102 SRPDB: The Signal Recognition Particle Database

Provider and access. The Signal Recognition Particle Database (SRPDB) is maintained at the University of Texas Health Science Center at Tyler, USA. It is accessible at: `http://psyche.uthct.edu/dbs/SRPDB/SRPDB.html`

Contents. SRPDB provides information about RNA, proteins, and receptors related to signal recognition particles. There are no links to external database.

Data modelling, data storage, and data acquisition. SRPDB is most likely a flat file database. Data annotation in SRPDB consists of, among others: sequence alignments (provided in various formats such as PDF, HTML, EMBL-format, etc.), pictures of secondary structure and three-dimensional models. All these data are most likely stored in files.

Data are acquired from external databases.

Data retrieval and query answering. SRPDB can be browsed by entering the data classification at various points. There are no query interfaces nor a possibility to download the data via FTP. In deed, the data items are represented as flat files thus they can be downloaded after browsing and retrieving.

Size. No information available.

Literature on SRPDB::

[SRPa] C. Zwieb and T. Samuelsson. SRPDB (Signal Recognition Particle Database). *Nucleic Acids Research*, 28(1):171-172, 2000.

A.103 SWISS-PROT

Provider and access. SWISS-PROT is available at: `http://www.expasy.ch/sprot/` and at: `http://www.ebi.ac.uk/swissprot/`
The database is currently an equal partnership project between the European Molecular Biology Laboratory (EMBL) with its outstation, the European Bioinformatics Institute (EBI) and the Swiss Institute of Bioinformatics (SIB).

```
ID   PILI_PSEAE     STANDARD;      PRT;   178 AA.
AC   P43502;
DT   01-NOV-1995 (Rel. 32, Created)
DT   01-NOV-1995 (Rel. 32, Last sequence update)
DT   30-MAY-2000 (Rel. 39, Last annotation update)
DE   PILI PROTEIN.
GN   PILI.
OS   Pseudomonas aeruginosa.
OC   Bacteria; Proteobacteria; gamma subdivision; Pseudomonadaceae;
OC   Pseudomonas.
RN   [1]
RP   SEQUENCE FROM N.A.
RC   STRAIN=ATCC 15692 / PAO1;
RX   MEDLINE=94195101; PUBMED=7908398;
RA   Darzins A.;
RT   "Characterization of a Pseudomonas aeruginosa gene cluster involved
RT   in pilus biosynthesis and twitching motility: sequence similarity to
RT   the chemotaxis proteins of enterics and the gliding bacterium
RT   Myxococcus xanthus.";
RL   Mol. Microbiol. 11:137-153(1994).
CC   -!- FUNCTION: MAY BE A PART OF A SIGNAL-TRANSDUCTION SYSTEM THAT
CC       REGULATES TWITCHING MOTILITY BY CONTROLLING PILUS FUNCTION
CC       (EXTENSION AND RETRACTION).
CC   -!- SIMILARITY: CONTAINS 1 CHEW DOMAIN.
CC   --------------------------------------------------------------------------
CC   This SWISS-PROT entry is copyright. It is produced through a collaboration
CC   between  the Swiss Institute of Bioinformatics  and the  EMBL outstation -
CC   the European Bioinformatics Institute.  There are no  restrictions on  its
CC   use  by  non-profit  institutions as long  as its content  is  in  no  way
CC   modified and this statement is not removed.  Usage  by  and for commercial
CC   entities requires a license agreement (See http://www.isb-sib.ch/announce/
CC   or send an email to license@isb-sib.ch).
CC   --------------------------------------------------------------------------
DR   EMBL; L22036; AAA25951.1; -.
DR   INTERPRO; IPR002545; -.
DR   PFAM; PF01584; CheW; 1.
KW   Sensory transduction.
SQ   SEQUENCE   178 AA;  19934 MW;  634A1A4B135A7E77 CRC64;
     MSDVQTPFQL LVDIDQRCRR LAAGLPAQQE AVQSWSGIGF RMGGRFFVAP MGEVGEVLHE
     PRYTQLPGVK TWVKGVANVR GRLLPIMDLC GFLGTELSPL RKQRRVLVVE HLDVFAGLIV
     DEVFGMQHFP VDTFSEQLPP LEAALQPFIH GVFHREQPWL VFSPHALAQH QGFLDVAV
//
```

FIGURE A.28: An example of a SWISS-PROT entry in raw text format ([SWISSc])

Contents. SWISS-PROT is a curated and annotated protein sequence database [SWISSa]. The database distinguishes itself from other protein sequence databases by four criteria:

- Annotation: The SWISS-PROT staff tries to include as much annotation information as possible. Annotation is done and periodically reviewed by the SWISS-PROT staff and external experts.

NicePro View of SWISS-PROT: P43502

[General] [Name and origin] [References] [Comments] [Cross-references] [Keywords] [Features] [Sequence] [Tools]

General information about the entry	
Entry name	**PILI_PSEAE**
Primary accession number	**P43502**
Secondary accession number(s)	None
Entered in SWISS-PROT in	Release 32, November 1995
Sequence was last modified in	Release 32, November 1995
Annotations were last modified in	Release 39, May 2000
Name and origin of the protein	
Protein name	PILI PROTEIN
Synonym(s)	None
Gene name(s)	PILI
From	Pseudomonas aeruginosa
Taxonomy	Bacteria; Proteobacteria; gamma subdivision; Pseudomonadaceae; Pseudomonas.

References

[1]
SEQUENCE FROM N.A.
STRAIN=ATCC 15692 / PAO1;
MEDLINE=94195101 [NCBI, ExPASy, Israel, Japan]; PubMed=7908398;
Darzins A.;
"Characterization of a Pseudomonas aeruginosa gene cluster involved in pilus biosynthesis and twitching motility: sequence similarity to the chemotaxis proteins of enterics and the gliding bacterium Myxococcus xanthus.";
Mol. Microbiol. 11:137-153(1994).

Comments

- ***FUNCTION***: MAY BE A PART OF A SIGNAL-TRANSDUCTION SYSTEM THAT REGULATES TWITCHING MOTILITY BY CONTROLLING PILUS FUNCTION (EXTENSION AND RETRACTION).
- ***SIMILARITY***: CONTAINS 1 CHEW DOMAIN.

Copyright

This SWISS-PROT entry is copyright. It is produced through a collaboration between the Swiss Institute of Bioinformatics and the EMBL outstation - the European Bioinformatics Institute. There are no restrictions on its use by non-profit institutions as long as its content is in no way modified and this statement is not removed. Usage by and for commercial entities requires a license agreement (See http://www.isb-sib.ch/announce/ or send an email to license@isb-sib.ch).

Cross-references	
EMBL	L22036; AAA25951.1; -. [EMBL / GenBank / DDBJ] [CoDingSequence]
INTERPRO	IPR002545; -.
PFAM	PF01584; CheW; 1.
PRODOM	[Domain structure / List of seq. sharing at least 1 domain]
BLOCKS	P43502.
DOMO	P43502.
PROTOMAP	P43502.
PRESAGE	P43502.
DIP	P43502.
SWISS-2DPAGE	GET REGION ON 2D PAGE.

FIGURE A.29: Part of a NiceProt view of the sample SWISS-PROT entry ([SWISSc])

- Minimal redundancy: Many sequence databases contain separate entries for a given protein sequence corresponding to different literature reports. The SWISS-PROT staff tries to merge these data to minimise redundancy. Conflicts between sequencing reports are indicated in the feature table line (see below) of the corresponding entry.

- Integration with other databases: SWISS-PROT is cross-referenced with currently over 30 different databases.

- Documentation: SWISS-PROT is distributed with a large number of index files and specialised documentation files.

SWISS-PROT provides links to currently over 30 other databases. The links are mostly hypertext links but can be sometimes simple textual references.

Data modelling, data storage, and data acquisition. The SWISS-PROT database is a flat file repository consisting of sequence entries. Sequence entries are composed of different line types, each with their own format. Each line type is beginning with a two-character line code. There are 22 different line types in SWISS-PROT. Some line types are optional or may occur more than once in an entry. For detailed information on each line type see [SWISSb] and [SWISSc]. Sequence entries consist of two different classes of data [SWISSc]:

- Core-data: consists of the sequence data, the citation information (bibliographical references) and the taxonomic data (i.e. the description of the biological source of the sequence).
- Annotation: consists of the description of items like function(s) of the protein, similarity to other proteins, secondary structure and many more. Annotation is mainly found in the comment (CC), feature table (FT) and keyword (KW) lines.

Each entry is labelled with an unique identifier. This is done by the (ID) line type which must be the first line of a regular SWISS-PROT entry. New data are acquired from TrEMBL (cf. A.104).

Data retrieval and query answering. The SWISS-PROT flat file is available for download via FTP at the following servers: `ftp.expasy.chandftp.ebi.ac.uk`
The file can also be obtained on CD-ROM at: `http://www.expasy.ch`

There are several ways to retrieve data from SWISS-PROT via the WWW interfaces at SIB and EBI. The SIB provides the Expert Protein Analysis System (ExPASy). Search can be performed as fixed-form queries against some (but not all!) special line type of the database entries (search by accession number, description, etc.) or against all fields of the database (full text search). Each entry as a result of a query can be viewed as plain text or by the *NiceProt* representation (i.e. a better readable HTML version of the entry). Examples are given in Figure A.28 and A.29. SWISS-PROT does not support a query-language thus there is no interface for *ad-hoc* queries.

The EBI provides the Sequence Retrieval System (SRS, cf. 4.3.7), an environment that integrates many databases and allows specialised *ad-hoc* and fixed-form queries.

There is no capability to browse the data of SWISS-PROT.

Size. SWISS-PROT contains currently over 88 000 sequence entries, comprising over 30 million amino acids abstracted from about 65 000 references [SWISSc]. The data file requires 185 MB and the documentation and index files require about 65 MB of disk space [SWISSa].

Literature on SWISS-PROT:

[SWISSa] A. Bairoch and R. Apweiler. The SWISS-PROT database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28(1):45-48, 2000.

[SWISSb] R. Apweiler. Introduction to Molecular Biology Databases. In R. Apweiler. The EBI Online Manual on Molecular Databases. 1999.

[SWISSc] `http://www.expasy.ch/sprot/sp-docu.html`

A.104 TrEMBL: The Supplement of SWISS-PROT

Provider and access. TrEMBL is maintained by the Swiss Institute of Bioinformatics (SIB) at Geneva/Switzerland in collaboration with the European Bioinformatics Institute (EBI) at Cambridge/UK. It is accessible via several sites, including among others the SIB and EBI sites at: `http://www.expasy.ch/sprot/sprot-top.html`
and at: `http://www.ebi.ac.uk/swissprot/`

Contents. One of the major goals of SWISS-PROT (cf. A.103) is to provide a maximum of annotation to the core data (protein sequences). Maintaining this high quality of annotation in SWISS-PROT requires careful sequence analysis. This is the rate-limiting step in the production of SWISS-PROT [SWISSa]. To address this concern, TrEMBL (Translation of EMBL nucleotide sequence database) was introduced in 1996 by the SWISS-PROT team. TrEMBL consists of computer-annotated entries derived from the translation of all coding sequences in the EMBL (cf. A.28) database.

Data modelling, data storage, and data acquisition. TrEMBL consists of two flat files representing the two main sections of the database:

- SP-TrEMBL (SWISS-PROT TrEMBL): contains all entries which should be incorporated into SWISS-PROT. SWISS-PROT accession numbers have been assigned to these entries. SP-TrEMBL is partially redundant against SWISS-PROT.
- REM-TrEMBL (REMaining TrEMBL): contains the entries which the SWISS-PROT annotators do not want to include in SWISS-PROT.

A typical TrEMBL entry looks very similar to a SWISS-PROT entry, since TrEMBL follows the SWISS-PROT format as closely as possible. The only (necessary) differences affecting the ID and DT line types (see [SWISSb] for detailed information).

TrEMBL acquire data only from the EMBL (cf. A.28) nucleotide sequence database.

Data retrieval and query answering. The TrEMBL data can be retrieved in the same way as the SWISS-PROT (cf. A.103) data.

Size. The zipped file with all TrEMBL data requires currently about 190 MB. TrEMBL is updated every week

Literature on TrEMBL:

[SWISSa] A. Bairoch and R. Apweiler. The SWISS-PROT database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28(1):45-48, 2000.

[SWISSb] R. Apweiler. Introduction to Molecular Biology Databases. In R. Apweiler. The EBI Online Manual on Molecular Databases. 1999.

A.105 SWISS-2DPAGE

Provider and access. SWISS-2DPAGE is maintained at the Swiss Institute of Bioinformatics (SIB) in collaboration with the Geneva University Hospital at Geneva, Switzerland. It is accessible through the SIB's server "Expasy" at: `http://www.expasy.ch/ch2d/`

Contents. SWISS-2DPAGE is an annotated two-dimensional gel electrophoresis database. It contains several reference maps (results of the gel electrophoresis). These reference maps have identified spots corresponding to separate protein entries in SWISS-2DPAGE.

Hypertext links to several other databases, such as MEDLINE (cf. A.59) and SWISS-PROT (cf A.103) are provided.

Data modelling, data storage, and data acquisition. The data are stored in flat files. The format of each entry is similar to the SWISS-PROT database (cf. A.103) and is described in [S2DPa] and [S2DPb]. There is no information about the source of data acquisition.

Data retrieval and query answering. The SIB homepage provides a fixed-form query interface. The reference maps can be browsed to retrieve information about the protein spots. The data files can also be downloaded via FTP. There is no *ad-hoc* query interface.

Size. The data are stored in three flat files requiring about 4 MB.

Literature on SWISS-2DPAGE:

[S2DPa] C. Hoogland et al. The 1999 SWISS-2DPAGE Database Update. *Nucleic Acids Research*, 28(1):286-288, 2000.

[S2DPb] `http://www.expasy.ch/ch2d/manch2d.html`

A.106 TAIR: The *Arabidopsis* Information Resource

Provider and access. The *Arabidopsis* Information Resource (TAIR) is a collaborative effort between the Carnegie Institution of Washington Department of Plant Biology and the National Center for Genome Resources (NCGR). The *Arabidopsis thaliana* Database (AtDB) and the An *Arabidopsis thaliana* Database (AAtDB) projects ended in Summer 1999. They both were succeeded by TAIR. The TAIR database will integrate both databases and will soon be accessible at: `http://www.arabidopsis.org/`

Contents. TAIR provides a genetic database with all information about the model organism *Arabidopsis thaliana* (thale cress). There are no information about cross-references to other databases, but we suggest that there will be several hypertext links to external databases.

Data modelling, data storage, and data acquisition. AtDB was based upon an object-model and implemented using the object DBMS Illustra (Informix). AAtDB was implemented using ACEDB as DBMS [TAIRa].

It is not yet documented how TAIR is modelled and implemented but there are some hints indicating that the data will be stored in an object DBMS and not using ACEDB.

Data are derived originally from the two databases as described above. New data are acquired from the research community. Special fill-in submission forms are provided.

Data retrieval and query answering. There is a fixed-form query interface which is still under construction. TAIR currently provides no *ad-hoc* interface. In the future also a flat file release will be provided for download via FTP.

There is no capability to browse the database for retrieval.

Size. The ace file with the entire AAtDB constructed by the ACEDB DBMS requires 74 MB (UNIX zipped TAR file). One should remind that this release is now out of date. Information about the size of TAIR was currently not available.

Literature on TAIR:

[TAIRa] `http://www.arabidopsis.org/release.html`

[TAIRb] `http://www.arabidopsis.org/TAIR.html`

A.107 The TIGR Databases

Provider and access. The TIGR Databases (TDB) are provided by The Institute for Genomic Research (TIGR) at Rockville, USA. The database can be accessed at: `http://www.tigr.org/tdb/`

Contents. The TIGR Databases provide are a collection of curated databases containing DNA and protein sequence, gene expression, cellular role, protein family, and taxonomic data for microbes, plants and humans. The collection consists of the following five databases:

- the TIGR *Arabidopsis thaliana* Database,
- the TIGR Rice Database,
- the TIGR Parasites Database,
- the TIGR Fungal Database, and
- the TIGR Gene Indices.

The TIGR Gene Indices are the primary service of TIGR. Hypertext links to GenBank (cf. A.36) are provided.

Data modelling, data storage, and data acquisition. The TIGR databases are managed by the relational DBMS Sybase. There is no information about the underlying database schema.

Data are acquired from GenBank (cf. A.36).

Data retrieval and query answering. Data can be accessed from TIGR via several fixed-form query interfaces. There is no *ad-hoc* query interface and no hierarchy to browse through.

In addition, the data can be downloaded as flat files via FTP (in TIGR format and FASTA [56] format).

Size. The current TIGR Gene Indices flat file in FASTA [56] format requires about 4 MB.

Literature on TIGR:

[TIGRa] J. Quackenbush et al. The TIGR Gene Indices: Reconstruction and Representation of expressed gene sequences. *Nucleic Acids Research*, 28(1):141-145, 2000.

[TIGRb] `http://www.tigr.org/tdb`

A.108 tmRDB: The tmRNA Database

Provider and access. The tmRNA Database (tmRDB) is maintained at the University of Texas Health Science Center at Tyler, USA in collaboration with the Auburn University in Auburn, USA. It can be accessed at: `http://psyche.uthct.edu/dbs/tmRDB/tmRDB.html`

Contents. tmRDB provides sequences of tmRNA a RNA molecule present in most bacteria and some organelles of eukaryotes. In addition, annotation to the sequence data is also provided.

tmRDB is cross-linked to GenBank (cf. A.36) via hypertext links.

Data modelling, data storage, and data acquisition. There is no information about the implementation of tmRDB. We assume that tmRDB is a flat file database.

Figure A.108 shows a sample entry of the database. The entries are represented as in Figure A.108. Each entry is explicitly structured using line types (search indexes).

Data are acquired from the scientific literature and other databases (primarily extractions from GenBank, cf. A.36).

Data retrieval and query answering. There are no query interfaces provided at the tmRDB homepage. Data can only be retrieved via browsing the classifications given by the developers.

```
ID   Chl.tra.
DE   Chlamydia trachomatis. 632 bases 1594C158 checksum
SQ             632 BP
     GGGGGUGua- aagguuu-cg aCUUAG-aAA ugaag--CGU U-----AAUU g---CAUGCG
     GAGGgc-GUU GGCUgg-CCU CCuaaaa--- AGCCGACaaa acaa---uaa augccgaacc
     uaaggcugaa ugcgaaauua uca------- --------gc uucgcugauc ucgaagaucu
     aagAGUAGCU gcuuaa---- ---------- ---------- ------uuag caaAGU-UGU
     Uaccuaaaua cgggugaccC GGUGU----- UCGcG--AGC Ucc------- --ACCAGA-G
     GUU------- ---------- UuCGAa---- ---ACACCGu ca------ug uaUCUGGUua
     g--------- ---------- ------AACU UAgguccUUU a-aUUCUCG- -AGGaaa--U
     GAGUUugaaa u--uuaaUGA GAGu---CGU UAGUCUCuau aGGg---GUU UcUAGCUga-
     gGAGACaUAA CGuauag--- ---------- ---------- ---------- ----------
     --uacCUAg- GAAC-uaaGC A-UGua-ga- -GGUU----- AGCGggga-- -guUUaCUAA
     GgacGAGAGu ucgacuCUCU CCACCUCCac ca
//
```

Figure A.30: Sample entry of tmRDB

There is no flat file release of tmRNA available for download via FTP.

Size. No information available.

Literature on tmRDB:

[tmRDBa] C. Zwieb and J. Wower. tmRDB (tmRNA Database). *Nucleic Acids Research*, 28(1):169-170, 2000.

[tmRDBb] `http://psyche.uthct.edu/dbs/tmRDB/AbouttmRDB.html`

A.109 TRANSFAC

Provider and access. TRANSFAC is developed and maintained at the Gesellschaft für Biotechnologisch Forschung (GBF) in Braunschweig, Germany and can be accessed at: `http://transfac.gbf.de/TRANSFAC/index.html`

Contents. TRANSFAC is a database on transcription factors. It is part of an environment at GBF providing four other databases (PathoDB, S/MARt DB, TRANSPATH, and CYTOMER). There are hypertext links to other external databases such as EMBL (cf. A.28), PIR (cf. A.79), and COMPEL (cf. A.15).

Data model, data storage, and data acquisition. TRANSFAC and the other integrated databases at GBF mentioned above are relational databases. The TRANSFAC data model consists of six major tables and over 90 other tables. A detailed description is available at [TFACb].

Data are acquired originally from the scientific literature.

Data retrieval and query answering. TRANSFAC offers a fixed-form query interface and a sort of *ad-hoc* query interface where the user can express regular expressions to formulate free-form queries. In addition, the data is classified into a hierarchy that can be browsed to retrieve single data items.

The entire database can be downloaded as flat files via FTP at: `ftp://transfac.gbf.de/pub/` (directory "transfac").

Note: TRANSFAC is also available (but not freely) as a Microsoft SQL-Server based relational database system.

Size. All (zipped) files together require approximately 2 MB.

Literature on TRANSFAC:

[TFACa] E. Wingender et al. TRANSFAC: an Integrated System for Gene Expression Regulation. *Nucleic Acids Research*, 28(1):316-319, 2000.

[TFACb] `http://transfac.gbf.de/TRANSFAC/doc/toc.html`

A.110 Transterm

Provider and access. Transterm, a database of messenger RNA components and signals, is maintained at the University of Otago, Dunedin/New Zealand. It can be accessed at: `http://biochem.otago.ac.nz/Transterm`

Contents. Transterm is a database of studies of messenger RNAs and translational control signals. The database focus on the untranslated regions (UTR) flanking a coding sequence (CDS). It seems that Transterm do not provide any interconnections with other databases.

Data modelling, data storage, and data acquisition. Transterm comprises a large number of files summarising sequence parameters, sequence datafiles and a relational database accessed via a WWW interface [TTa]. There is no information about the relational model. The flat files are structured in several different formats.

Data are derived only from GenBank (cf. A.36).

Data retrieval and query answering. There are several fixed-form query interface to retrieve data from Transterm. Data files can also be downloaded via FTP.

There is no *ad-hoc* query interface. The data cannot be browsed through.

Size. Transterm provides many different data files each with a different size. Central information about the current size of the files or the database is not available.

Literature on Transterm:

[TTa] G. Jacobs et al. Transterm: a Database of Messenger RNA Components and Signals. *Nucleic Acids Research*, 28(1):293-295, 2000.

[TTb] `http://biochem.otago.ac.nz/Transterm`

A.111 TRIPLES: Database of TRansposon-Insertion Phenotypes, Localisation, and Expression in Saccharomyces

Provider and access. TRIPLES is maintained by the Yale Genome Center at New Haven/USA and can be accessed freely at: `http://ygac.med.yale.edu`

Contents. TRIPLES is a genetic database for the model organism *S. cerevisiae*. The focus of TRIPLES is collecting data on defined mutant alleles, for the analysis of disruption phenotypes, protein localisation, and gene expression. Hypertext links are provided to the SGD (cf. A.100), YPD (cf. A.119), and GenBank (cf. A.36).

Data modelling, data storage, and data acquisition. TRIPLES is implemented using the relational DBMS Oracle. There is no information about the underlying data model. The ODBC (Open database Connectivity) was used to implement the database access. This ensures code compatibility with different database platforms.

TRIPLES houses both a private and public area. Members of the Yale Genome Analysis Center may enter a password-protected area of the database to upload new data.

Data are probably acquired from the research community.

Data retrieval and query answering. The user can choose between three types of data to search for:

- disruption phenotypes
- protein localisation
- gene expression

Each category provides a fixed-form query interface that allows searching by gene name or clone ID. Results can be viewed in HTML-table format. Data can also be downloaded after querying in tab-delimited text files. There are no *ad-hoc* query interfaces nor any capabilities to browse the data.

Size. No information available.

Literature on TRIPLES:

[TRI] A.K. Kumar et al. TRIPLES: a Database of Gene Function in S. cerevisiae. *Nucleic Acids Research*, 28(1):81-84, 2000.

A.112 TRRD: The Transcription Regulatory Regions Database

Provider and access. The Transcription Regulatory Regions Database (TRRD) is maintained by a collaboration of several parts of the Siberian Branch of the Russian Academy of Science at Novosibirsk, Russia. It is accessible at: `http://wwwmgs.bionet.nsc.ru/mgs/dbases/trrd4/`

Contents. TRRD is designed for accumulation of experimental data on extended regulatory regions of eukaryotic genes.

Hypertext links to other external databases such as SWISS-PROT (cf. A.103), EMBL (cf. A.28), GenBank (cf. A.36), and TRANSFAC (cf. A.109) are provided.

Data modelling, data storage, and data acquisition. The TRRD seems to be implemented as a flat file format. The entries are structured explicitly trough line types (search indexes). Currently there are more than 80 different line types allowed in an TRRD entry.

There is no information available about data acquisition.

Data retrieval and query answering. TRRD can only be browsed to retrieve data items. The data is classified into a hierarchy. There are no query forms at all nor is there a possibility to download the data as flat file(s).

In addition, the TRRD database can be accessed via SRS at several servers, including that at the TRRD homepage.

Size. There is no information available.

Literature on TRRD:

[TRRDa] N. Kolchanov et al. Transcription Regulatory Regions Database (TRRD): Its Status in 2000. *Nucleic Acids Research*, 28(1):298-301, 2000.

[TRRDb] `http://wwwmgs.bionet.nsc.ru/mgs/dbases/trrd4/trrdintro.html`

A.113 UK CropNet: The UK Crop Plant Bioinformatics Network

Provider and access. The UK Crop Plant Bioinformatics Network (UK CropNet) maintains eight databases. It is accessible at: `http://ukcrop.net/`.

Contents. UK CropNet provides eight genetic databases:

- the *Arabidopsis* Genome Resource (AGR)
- BarleyDB
- BrassicaDB
- CerealsDB
- Comparative Mapping (ComapDB)
- Forage Grasses (FoggDB)
- MilletGenes
- SpudBase

Further information about the contents of these databases can be retrieved in [CROPa].

Each database provides hypertext links to external databases.

Data modelling, data storage, and data acquisition. All databases provided by UK CropNet are modelled and implemented using ACEDB as DBMS. The schema of each of the eight and several other plant databases (also implemented using ACEDB) can be browsed with the ACEDB class browsing tool WebAce at [CROPb].

Data are acquired primarily from the plant research community.

Data retrieval and query answering. The different databases at UK CropNet can be browsed or searched via the tools provided by ACEDB. This includes fixed-form and *ad-hoc* query interfaces.

The ACEDB software package can be downloaded for local installation on several platforms. All databases provide flat file releases for a local ACEDB system. These files can also be downloaded via FTP.

Size. No information available.

Literature on UK CropNet:

[CROPa] J. Dicks et al. K CropNet: a Collection of Databases and Bioinformatics Resources for Crop Plant Genomics. *Nucleic Acids Research*, 28(1):104-107, 2000.

[CROPb] `http://ukcrop.net/db.html`

A.114 UTRdb

Provider and access. UTRdb is maintained by a collaboration of several italian and american research institutes and universities. It is provided at CNR, Bari, Italy and can be accessed at: `http://bigarea.area.ba.cnr.it:8000/EmbIT/UTRHome/`

Contents. UTRdb is an annotated sequence database about untranslated regions of eukaryotic mRNAs.

It provides Hypertext links to external databases.

Data modelling, data storage, and data acquisition. UTRdb is a flat file database consisting of entries. Each entry is explicitly structured by search indexes. A sample entry of URTdb can be seen in [UTRa].

Data is acquired from EMBL (cf. A.28).

Data retrieval and query answering. UTRdb can only be accessed via the Sequence Retrieval System (SRS, cf. Chapter 4.3.7). A local SRS server is accessible at: `http://bighost.area.ba.cnr.it/srs/`

There are no query interfaces, nor any lat file released for downloading, nor a capability to browse through the data.

Size. No information available.

Literature on UTRdb:

[UTRa] G. Pesole et al. UTRdb and UTRsite: specialized databases of sequences and functional elements of 5' and 3' untranslated regions of eukaryotic mRNAs. *Nucleic Acids Research*, 28(1):193-196, 2000.

A.115 WIT: The What Is There System

Provider and access. The What Is There (WIT) system is maintained by a collaboration between Integrated Genomics at Chicago/USA and Argonne National Laboratory at Argonne/USA. It can be accessed at the web site of both collaborators: `http://wit.mcs.anl.gov/WIT2/`
and at: `http://wit.IntegratedGenomics.com/IGwit`

Contents. The database supports comparative analysis of sequenced genomes and allows generation of metabolic reconstructions based on data from the EMP/MPW (cf. A.69) family of databases. WIT provides hypertext links links to MPW (cf. A.69) and MEDLINE (cf A.59).

Data modelling, data storage, and data acquisition. There are no information available about the data model and the implementation.

WIT acquires chromosomal sequences and metabolic pathways from the MPW (cf. A.69).

Data retrieval and query answering. Data in WIT can only be accessed via fixed-form query interfaces. WIT can be searched for genes, proteins, COGs, metabolic pathways, etc.

There is no *ad-hoc* query interface, no capability to browse the data, nor any flat file release provided for download.

A large set of analysis tools are within the database. The results of any query can be analysed using these programs.

Size. No information available.

Literature on WIT:

[WITa] R. Overbeek et al. WIT: Integrated System for High-Throughput Genome Sequence Analysis and Metabolic Reconstruction. *Nucleic Acids Research*, 28(1):123-125, 2000.

[WITb] `http://wit.mcs.anl.gov/WIT2/`

A.116 WormPD: The *C.elegans* Proteome Database

Provider and access. WormPD is maintained by Proteome Inc. at Beverly/USA and accessible freely to academic users at: `http://www.proteome.com/databases/index.html`

Contents. The WormPD stores proteome information about the proteins of the model organism *C.elegans*. Hypertext links to several other databases such as e.g. GenBank (cf. A.36), SWISS-PROT (cf. A.103), MIPS (cf. A.63), and PubMed (cf. A.90) are provided.

Data modelling, data storage, and data acquisition. WormPD is based on a relational model and uses the commercial software Oracle as DBMS. Further information about the data model is not provided. A controlled vocabulary for querying and analysing data in WormPD is enforced.

The data are gathered from the scientific literature and submissions of the community.

Data retrieval and query answering. The data are accessible via a (complex) fixed-form query interfaces or an advanced full-text keyword search (several keywords can be combined by using the boolean operators "AND"/"OR"). WormPD can also be searched by the gene name encoding a protein. The database can also be browsed starting at [WorDb], following links classified by several categories to at least a list of all data items in the according category.

The data items are represented in a special format called the "Protein Report". The Protein Report lists all information about a protein in a HTML table. A comprehensive reference list with links to their MEDLINE (cf. A.59) abstracts are provided.

There are no flat files available. There is also no *ad-hoc* query interface.

Size. The current release (November 2000) stores information about 18973 proteins and provides 2415 references. The size of the data in MB is not available.

Literature on WormPD:

[WorDa] M. Costanzo et al. The Yeast Proteome Database and Caenorhabditis elegans Proteome Database: comprehensive resources for organisation and comparison of model organism protein information. *Nucleic Acids Research*, 28(1):73-76, 2000.

[WorDb] `http://www.proteome.com/databases/index.html`

A.117 XREFdb

Provider and access. The XREFdb database was maintained at the US National Center for Biotechnology Information (NCBI) in collaboration with the John Hopkins University School of Medicine in Baltimore, USA, and the University of British Columbia at Vancouver, Canada. It was able to be accessed at: `http://www.ncbi.nlm.nih.gov/XREFdb/`

The XREF project has been completed and all XREF software is no longer supported.

A.118 YIDB: The Yeast Intron DataBase

Provider and access. The Yeast Intron DataBase (YIDB) is free for academic users and can be accessed at: `http://www.EMBL-Heidelberg.DE/ExternalInfo/seraphin/yidb.html`.

Contents. YIDB contains currently available information about all introns encoded in the nuclear and mitochondrial genomes of *S. cerevisiae*.

The introns are divided according to their mechanism of excision into five classes:

- group I introns
- group II introns
- pre-mRNA introns
- tRNA introns
- the HAC1 intron

For a detailed description of these five classes see [YIDBb]

Hypertext links are provided to EMBL (cf A.28), MIPS (cf. A.63), SGD (A.100), and PubMed (cf. A.90).

Data modelling, data storage, and data acquisition. There are no information available about the underlying data model and data management system. We assume that the data are stored in five files, each of it structured as (HTML) tables.

The data are derived from other databases such as e.g. SGD (cf. A.100), YPD (cf. A.119), MIPS (cf. A.63), and EMBL (cf. A.28) and complemented with information (manually) retrieved from the literature.

Data retrieval and query answering. The YIDB can be browsed following the link to one of the five intron classes. All data items are listed in an HTML table, representing the information that is available and providing hypertext links to the databases mentioned above.

There are no query interfaces at all nor a flat file release provided for download.

Size. No information available.

Literature on YIDB:

[YIDBa] P. Lopez and B. Seraphin. YIDB: the Yeast Intron DataBase. *Nucleic Acids Research*, 28(1):85-86, 2000.

[YIDBb] `http://www.EMBL-Heidelberg.DE/ExternalInfo/seraphin/yidb.html`

A.119 YPD: The Yeast Proteome Database

Provider and access. The Yeast Proteome Database (YPD) is maintained by Proteome Inc. at Beverly/USA and accessible freely to academic users at: `http://www.proteome.com/databases/index.html`

Contents. The YPD stores proteome information about the proteins of the model organism *S. cerevisiae*.

Hypertext links to several other databases such as e.g. GenBank (cf. A.36), SWISS-PROT (cf. A.103), MIPS (cf. A.63), and PubMed(cf. A.90) are provided.

Data modelling, data storage, and data acquisition. YPD is based on a relational model and uses the commercial software Oracle as DBMS. Further information about the data model is not provided. A controlled vocabulary for querying and analysing data in YPD is enforced.

The data are gathered from the scientific literature and submissions of the community.

Data retrieval and query answering. The data are accessible via a (complex) fixed-form query interfaces or an advanced full-text keyword search (several keywords can be combined by using the boolean operators "AND"/OR"). YPD can also be searched by the gene name encoding a protein. The database can also be browsed starting at [YPDb], following links classified by several categories to at least a list of all data items in the according category.

The data items are represented in a special format called the "Protein Report". The Protein Report lists all information about a protein in a HTML table. A comprehensive reference list with links to their MEDLINE (cf. A.59) abstracts are provided.

There is no flat file release available.

Size. The current release (November 2000) stores information about 6145 proteins and provides 18875 references. The size of the data in MB is not available.

Literature on YPD:

[YPDa] M. Costanzo et al. The Yeast Proteome Database and Caenorhabditis elegans Proteome Database: comprehensive resources for organisation and comparison of model organism protein information. *Nucleic Acids Research*, 28(1):73-76, 2000.

[YPDb] `http://www.proteome.com/databases/index.html`

A.120 ZmDB: *Zea mays* DataBase

Provider and access. The *Zea mays* DataBase (ZmDB) is maintained at the Iowa State University at Ames, USA and can be accessed at: `http://zmdb.iastate.edu/`

Contents. ZmDB stores sequence, expression, and phenotype data of the major crop plant maize.

Entries in ZmDB are hypertext linked to the data resources at NCBI such as e.g. GenBank (cf. A.36), OMIM (cf. A.72), and PubMed (cf. A.90).

Data modelling, data storage, and data acquisition. ZmDB is implemented using the ACEDB software as DBMS. There is no information about the underlying schema, but the existing classes can be browsed with the query tool "Class Browser" (see below) at [ZmDBb].

Data are acquired from other public databases and the scientific community.

Data retrieval and query answering. Data in ZmDB can be accessed as follows:

1. Fixed-form query interfaces:

- Text Search: find by content, search for objects with a specific text string
- Blast Search: search for homologs of given sequences in the database using BLAST [4].

2. *Ad-hoc* query interfaces:

   - Queries can be expressed in the AQL.

In addition the data can be downloaded via FTP as flat files, as well as several analysis tools.

A tool called "Class Browser" supports browsing through the database class by class.

Size. The flat files of ZmDB currently store between about 6 MB and 150 MB.

Literature on :

[ZmDBa] X. Gai et al. Gene Discovery Using the Maize Genome Database ZmDB. *Nucleic Acids Research*, 28(1):94-96, 2000.

[ZmDBb] `http://zmdb.iastate.edu/`

Appendix B

The Genetic Code

B.1 Index of Nucleotides

The following table lists all nucleotides that occur in nucleic acids (DNA and RNA).

TABLE B.1: Index of nucleotides that occur in DNA and RNA.

Chemical name	One-letter code	Occurs in
Adenin	A	DNA and RNA
Cytosin	C	DNA and RNA
Guanin	G	DNA and RNA
Thymin	T	DNA only
Uracil	U	RNA only

B.2 Index of Amino Acids

The following table (Table B.2) provides a list of the one- and three-letter code of the 20 different amino acids that occur in proteins. The column named "classified as" provides information about the chemical properties of the residue of the according amino acid. The classification is according to [9] and distinguishes between: Hydrophobic amino acids, charged amino acids, polar amino acids, and Glycine. (The amino acid Glycine is an exception. It cannot be put into one of the classes mentioned before.) A list of the chemical formulas of all 20 amino acids can be seen e.g. in [45] or [9].

TABLE B.2: Index of the 20 aminoacids that occur in living organisms.

One-letter code	Three-letter code	Chemical name	Classified as
A	Ala	Alanine	hydrophobic
B	Asx	either of D or N	cf. D or N
C	Cys	Cysteine	polar
D	Asp	Aspartic acid	charged
E	Glu	Glutamatic acid	charged
F	Phe	Phenylalanine	hydrophobic
G	Gly	Glycerine	Glycerine
H	His	Histidine	polar
I	Ile	Isoleucine	hydrophobic
K	Lys	Lysine	charged
L	Leu	Leucine	hydrophobic
M	Met	Methionine	hydrophobic
N	Asn	Asparagine	polar
P	Pro	Proline	hydrophobic
Q	Gln	Glutamine	polar
R	Arg	Arginine	charged
S	Ser	Serine	polar
T	Thr	Threonine	polar
V	Val	Valine	hydrophobic
W	Trp	Tryptophan	polar
X	X	undetermined amino acid	–
Y	Tyr	Tyrosine	polar
Z	Glx	either of E or Q	cf. E or Q

B.3 The Genetic Code

Table B.3 shows the "universal genetic code", which determines the result of each mRNA codon. The genetic information is transfered from genes (DNA) into mRNA according the base-pairing rules. After that, the sequence of nucleotides is translated into a sequence of aminoacids (protein). Three nucleotides in mRNA code for one amino acid in the protein. Abbreviations of the nucleotides and amino acids are explained in B.1 and B.2.

Example: The codon consiting of UCG codes for the amino acid serine (Ser). "STOP" means the termination of the translation procedure.

TABLE B.3: The genetic code

First Base	**Second Base**				**Third Base**
	U	*C*	*A*	*G*	
U	Phe	Ser	Tyr	Cys	*U*
	Phe	Ser	Tyr	Cys	*C*
	Leu	Ser	STOP	STOP	*A*
	Leu	Ser	STOP	Trp	*G*
C	Leu	Pro	His	Arg	*U*
	Leu	Pro	His	Arg	*C*
	Leu	Pro	Gln	Arg	*A*
	Leu	Pro	Gln	Arg	*G*
A	Ile	Thr	Asn	Ser	*U*
	Ile	Thr	Asn	Ser	*C*
	Ile	Thr	Lys	Arg	*A*
	Met (START)	Thr	Lys	Arg	*G*
G	Val	Ala	Asp	Gly	*U*
	Val	Ala	Asp	Gly	*C*
	Val	Ala	Glu	Gly	*A*
	Val	Ala	Glu	Gly	*G*

Bibliography

[1] Web Tutorial on DNA and Molecular Genetics. http://gened.emc.maricopa.edu/bio/bio181/BIOBK/BioBookDNAMOLGEN.html.

[2] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann Publishers, 2000.

[3] http://genome.cornell.edu/acedocs/.

[4] S. Altschul et al. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[5] http://asn1.elibel.tm.fr/.

[6] P. Baker et al. Tambis – transparent access to multiple bioinformatics information sources. In *Proceedings of the 6th International Conference on Intelligent Systems in Molecular Biology*, 1998.

[7] P. Baker et al. An ontology for bioinformatics applications. *Bioinformatics*, 15(6):510–520, 1999.

[8] G. Booch, J. Rumbaugh, and I. Jacobson. The Unified Modeling Language (UML). specification – version 1.0. UML Web Site: http://www.rational.com/uml, January 1997.

[9] C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland Publishing, Inc., 1991.

[10] V. Breazu-Tannen, P. Buneman, and L. Wong. Naturally embedded query languages. In J. Biskup and R. Hull, editors, *Proceedings of the 4th International Conference on Database Theory*, volume 646 of *Lecture Notes in Computer Science (LNCS)*, pages 140–154. Springer Verlag, 1992.

[11] T.A. Brown. *Gene Cloning*. Spektrum Akademischer Verlag GmbH, second edition, 1993. (in German).

[12] P. Buneman et al. Principles of programming with complex objects and collection types. *Theoretical Computer Science*, 149(3), 1995.

[13] P. R. Butler. Modelling metabolism and metabolic control analysis. available at: http://www.bi.umist.ac.uk/users/mjfprb/2IRM/default.asp. Online material of a teaching course at the UMIST (University of Science and Technology in Manchester).

[14] P. Carter et al. SRS: Analysing and using data from heterogeneous textual databanks. In S. Letovsky, editor, *Bioinformatics: Databases and Systems*, chapter 18, pages 213–231. Kluwer Academic Publishers, Boston, Dordrecht, London, second edition, 2000.

[15] R. Cattell, editor. *The Object Database Standard: ODMG-93*. Morgan Kaufmann Publishers, 1993.

[16] I-M. Chen and V. Markowitz. An overview of the object protocol model (OPM) and the OPM data management tools. *Information Systems*, 20(5):393–418, 1995.

[17] T. Chen, H. He, and G Church. Modeling gene expression with differential equations. In *Pacific Symposium on Biocomputing*, pages 29–40, 1999.

[18] P. Clote and R. Backofen. *Computational Molecular Biology, an Introduction*. John Wiley & Sons, Ltd., Chichester, New York, Weinheim, Brisbane, Singapore, and Toronto, 2000.

[19] S. Conrad. *Federated Databasesystems: Concepts of Data Integration*. Springer-Verlag, 1997. (in German).

[20] S.B. Davidson, C. Overton, V. Tannen, and L. Wong. Biokleisli: A digital library for biomedical researchers. *International Journal an Digital Libraries*,, 1(1):36–53, 1997.

[21] H. de Jong. Modelling and simulation of genetic regulatory systems: A literature review. Research report, INRIA, 2000. available via FTP at: `ftp.inria.fr`.

[22] `http://www.w3.org/DOM/`.

[23] R. Durbin and J. Thierry-Mieg. The ACEDB genome database. available at: `http://genome.cornell.edu/acedocs/`.

[24] T. Etzold, A. Ulyanow, and P. Argos. SRS: Information retrieval system for molecular biology data banks. *Methods in Enzymology*,, 266:114–128, 1996.

[25] N. Friedman et al. Using bayesian networks to analyze expression data. In *Proceedings of the 4th Annual Iternational Conference on Computational Molecular Biology (RECOMB)*. ACM Press, 2000.

[26] W. Fujibuchi et al. DBGET/LinkDB: an integrated database retrieval system. In *Pacific Symposium on Biocomputing*, pages 683–694, 1997.

[27] `http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html`.

[28] D. George, H.W. Mewes, and H. Kihara. A standardized format for sequence data exchange. *Protein Sequence Data Analysis*, 1:27–39, 1987.

[29] A. Gombert and J. Nielsen. Mathematical modelling of metabolism. *Current Opinion in Biotechnology*, 11:180–186, 2000.

[30] S. Goto et al. Organizing and computing metabolic pathway data in terms of binary relations. In *Pacific Symposium on Biocomputing*, pages 175–186, 1997.

[31] Object Management Group. CORBA: Architecture and specification. OMG publication, 1996.

[32] S Hall and Spadaccini. The STAR file: Detailed specifications. *Journal of Chemical Information and Computer Sciences*, 34:505–508, 1994.

[33] K. Heidtke and S. Schulze-Kremer. Design and implementation of a qualitative simulation model of λ phage infection. *Bioinformatics*, 14(1):81–91, 1998.

[34] R. Hofestädt and F. Meinecke. Interactive modelling and simulation of biochemical networks. *Computers in Biology and Medicine*, 25(3):321–334, 1995.

[35] R. Hofestädt and U. Scholz. Information processing for the analysis of metabolic pathways and inborn errors. *Biosystems*, 47:91–102, 1998.

[36] Interpreter of commands and recursive syntax. In: The SRS user manual. available at: `http://srs.ebi.ac.uk:5000/man/srsman.html`.

[37] Linge J., M. Nilges, and L. Ehrlich. StarDOM: From STAR format to XML. *Journal of Biomolecular NMR*, 15:169–172, 1999.

[38] K. Jungfer, G. Cameron, and T. Flores. EBI: CORBA and the EBI databases. In S. Letovsky, editor, *Bioinformatics: Databases and Systems*, chapter 20, pages 245–254. Kluwer Academic Publishers, Boston, Dordrecht, London, second edition, 2000.

[39] P. Karp. A strategy for database interoperation. *Journal of Computational Biology*, 2(4):573–586, 1995.

[40] P. Karp. Metabolic databases. *Trends in Biochemistry Sciences*, 23:114–116, 1998.

[41] P. Karp and M. Mavrovouniotis. Representing, analysing, and synthesizing biochemical pathways. *IEEE Expert*, 9(2):11–21, 1994.

[42] P. Karp and S. Paley. Representation of metabolic knowledge: Pathways. In *Pacific Symposium on Biocomputing*, pages 203–211, 1994.

[43] H. Kitakami et al. Building and search system for a large-scale DNA database. In S. Schulze-Kremer, editor, *Advances in Molecular Bioinformatics*, pages 123–138. ISO Press, 1994.

[44] K. Kyoda, M. Muraki, and H. Kitano. Construction of a generalized simulator for multi-cellular organisms and its application to SMAD signal transduction. In *Pacific Symposium on Biocomputing*, pages 317–328, 2000.

[45] A. Lehninger, D. Nelson, and M. Cox. *Principles of Biochemistry*. Spektrum Akademischer Verlag GmbH, Heidelberg, Berlin, Oxford, second edition, 1998. (in German).

[46] S. Letovsky, editor. *Bioinformatics: Databases and Systems*. Kluwer Academic Publishers, Boston, Dordrecht, London, second edition, 2000.

[47] V. Markowitz et al. Facilities for exploring molecular biology databases on the web: A comparative study. In *Pacific Symposium on Biocomputing*, pages 256–267, 1997.

[48] V. Markowitz et al. OPM:object-protocol model data management tools '97. In S. Letovsky, editor, *Bioinformatics: Databases and Systems*, chapter 16, pages 187–200. Kluwer Academic Publishers, Boston, Dordrecht, London, second edition, 2000.

[49] H. Matsuno et al. Hybrid petri net representation of gene regulatory network. In *Pacific Symposium on Biocomputing*, pages 338–349, 2000.

[50] H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269:650–656, 1995.

[51] R. McEntire et al. An evaluation of ontology exchange languages for bioinformatics. In *Proceedings of the 8th International Conference on Intelligent Systems in Molecular Biology*, 2000.

[52] P. Mendes. GEPASI: A software package for modelling the dynamics, steady states and control of biochemical and other systems. *Computer Applications in the Biosciences*, 9(5):563–571, 1993.

[53] G. Michal, editor. *Biochemical Pathways*. Spektrum Akademischer Verlag GmbH, Heidelberg, Berlin, 1999. (in German).

[54] R. Milner et al. *The Definition of Standard ML*. MIT Press, 1990.

[55] N. Paton et al. Query processing in the TAMBIS bioinformatics source integration system. In *Proceedings of the 11th Conference on Scientific and Statistical Databases (SSDBM)*, pages 138–147. IEEE Press, 1999.

[56] W. Pearson and D. Lipman. Improved tools for biological sequence comparison. In *Proceedings of the National Academy of Science USA (PNAS)*, volume 85, pages 2444–2448, 1988.

[57] F. Rechenmann. Knowledge bases and computational molecular biology. In N. Mars, editor, *Towards Very Large Knowledge Bases*, pages 7–12. IOS Press, 1995.

[58] A. Rector et al. The GALEN modelling language for medical terminology. *AI in Medicine*, 9:139–171, 1997.

[59] V. Reddy, M. Mavrovouniotis, and M. Liebman. Petri net representation in metabolic pathways. In *Proceedings of the 1st International Conference on Intelligent Systems in Molecular Biology*, 1993.

[60] R. Sayle and E. Milner-White. RasMol: Biomolecular graphics for all. *Trends in Biochemical Sciences*, 20(9):374, 1995.

[61] J. Schaff and L. Loew. The virtual cell. In *Pacific Symposium on Biocomputing*, pages 228–239, 1999.

[62] O. Schmelzer et al. Building large knowledge bases in molecular biology. In *Proceedings of the 1st International Conference on Intelligent Systems in Molecular Biology*, 1993.

[63] S. Schulze-Kremer. Ontologies for molecular biology. In *Pacific Symposium on Biocomputing*, pages 693–707, 1998.

[64] J. Siegel. *CORBA Fundamentals and Programming*. John Wiley and Sons, New York, 1996.

[65] K. Sirotkin. NCBI: Integrated data for molecular biology research. In S. Letovsky, editor, *Bioinformatics: Databases and Systems*, chapter 1, pages 12–19. Kluwer Academic Publishers, Boston, Dordrecht, London, second edition, 2000.

[66] P. Smolen, D. Baxter, and J. Byrne. Modeling transcriptional control in gene networks – methods, recent results, and future directions. *Bulletin of Mathematical Biology*, 62:247–292, 2000.

[67] A. Stokes, H. Matsuda, and A. Hashimoto. GXML: A novel method for exchanging and querying complete genomes by representing them as structured documents. *Transactions of Information Processing Society of Japan*, 40(3):66–78, 1999. available at: `http://yebisu.ics.es.osaka-u.ac.jp/~matsuda/e-publish.html`.

[68] M. Tomita et al. E-CELL: Software environment for whole-cell simulation. *Bioinformatics*, 15(1):72–84, 1999.

[69] S. Tsur. Data mining in the bioinformatics domain. In *Proceedings of the 26th Conference on Very Large Databases (VLDB)*, 2000.

[70] R. Unwin et al. Biology Workbench: a computing and analysis environment for the biological sciences. In S. Letovsky, editor, *Bioinformatics: Databases and Systems*, chapter 19, pages 233–244. Kluwer Academic Publishers, Boston, Dordrecht, London, second edition, 2000.

[71] J. van Helden et al. Representing and analysing molecular and cellular function using the computer. *Biological Chemistry*, 381:921–935, 2000.

[72] G. Xie, R. DeMarco, R. Blevins, and Y. Wang. Storing biological sequence databases in relational form. *Bioinformatics*, 16(3):288–289, 2000.

[73] `http://www.w3.org/TR/xlink/`.

[74] `http://www.w3.org/TR/xquery/`.