SECTION ONE

Introduction and Biological Databases

CHAPTER ONE

Introduction

Quantitation and quantitative tools are indispensable in modern biology. Most biological research involves application of some type of mathematical, statistical, or computational tools to help synthesize recorded data and integrate various types of information in the process of answering a particular biological question. For example, enumeration and statistics are required for assessing everyday laboratory experiments, such as making serial dilutions of a solution or counting bacterial colonies, phage plaques, or trees and animals in the natural environment. A classic example in the history of genetics is by Gregor Mendel and Thomas Morgan, who, by simply counting genetic variations of plants and fruit flies, were able to discover the principles of genetic inheritance. More dedicated use of quantitative tools may involve using calculus to predict the growth rate of a human population or to establish a kinetic model for enzyme catalysis. For very sophisticated uses of quantitative tools, one may find application of the "game theory" to model animal behavior and evolution, or the use of millions of nonlinear partial differential equations to model cardiac blood flow. Whether the application is simple or complex, subtle or explicit, it is clear that mathematical and computational tools have become an integral part of modern-day biological research. However, none of these examples of quantitative tool use in biology could be considered to be part of bioinformatics, which is also quantitative in nature. To help the reader understand the difference between bioinformatics and other elements of quantitative biology, we provide a detailed explanation of what is bioinformatics in the following sections.

Bioinformatics, which will be more clearly defined below, is the discipline of quantitative analysis of information relating to biological macromolecules with the aid of computers. The development of bioinformatics as a field is the result of advances in both molecular biology and computer science over the past 30–40 years. Although these developments are not described in detail here, understanding the history of this discipline is helpful in obtaining a broader insight into current bioinformatics research. A succinct chronological summary of the landmark events that have had major impacts on the development of bioinformatics is presented here to provide context.

The earliest bioinformatics efforts can be traced back to the 1960s, although the word *bioinformatics* did not exist then. Probably, the first major bioinformatics project was undertaken by Margaret Dayhoff in 1965, who developed a first protein sequence database called *Atlas of Protein Sequence and Structure*. Subsequently, in the early 1970s, the Brookhaven National Laboratory established the Protein Data Bank for archiving three-dimensional protein structures. At its onset, the database stored less

than a dozen protein structures, compared to more than 30,000 structures today. The first sequence alignment algorithm was developed by Needleman and Wunsch in 1970. This was a fundamental step in the development of the field of bioinformatics, which paved the way for the routine sequence comparisons and database searching practiced by modern biologists. The first protein structure prediction algorithm was developed by Chou and Fasman in 1974. Though it is rather rudimentary by today's standard, it pioneered a series of developments in protein structure prediction. The 1980s saw the establishment of GenBank and the development of fast database searching algorithms such as FASTA by William Pearson and BLAST by Stephen Altschul and coworkers. The start of the human genome project in the late 1980s provided a major boost for the development of bioinformatics. The development and the increasingly widespread use of the Internet in the 1990s made instant access to, and exchange and dissemination of, biological data possible.

These are only the major milestones in the establishment of this new field. The fundamental reason that bioinformatics gained prominence as a discipline was the advancement of genome studies that produced unprecedented amounts of biological data. The explosion of genomic sequence information generated a sudden demand for efficient computational tools to manage and analyze the data. The development of these computational tools depended on knowledge generated from a wide range of disciplines including mathematics, statistics, computer science, information technology, and molecular biology. The merger of these disciplines created an information-oriented field in biology, which is now known as *bioinformatics*.

WHAT IS BIOINFORMATICS?

Bioinformatics is an interdisciplinary research area at the interface between computer science and biological science. A variety of definitions exist in the literature and on the world wide web; some are more inclusive than others. Here, we adopt the definition proposed by Luscombe et al. in defining bioinformatics as a union of biology and informatics: *bioinformatics* involves the technology that uses computers for storage, retrieval, manipulation, and distribution of information related to biological macromolecules such as DNA, RNA, and proteins. The emphasis here is on the use of computers because most of the tasks in genomic data analysis are highly repetitive or mathematically complex. The use of computers is absolutely indispensable in mining genomes for information gathering and knowledge building.

Bioinformatics differs from a related field known as *computational biology*. Bioinformatics is limited to sequence, structural, and functional analysis of genes and genomes and their corresponding products and is often considered *computational molecular biology*. However, computational biology encompasses all biological areas that involve computation. For example, mathematical modeling of ecosystems, population dynamics, application of the game theory in behavioral studies, and phylogenetic construction using fossil records all employ computational tools, but do not necessarily involve biological macromolecules.

Beside this distinction, it is worth noting that there are other views of how the two terms relate. For example, one version defines *bioinformatics* as the development and application of computational tools in managing *all kinds* of biological data, whereas *computational biology* is more confined to the theoretical development of algorithms used for bioinformatics. The confusion at present over definition may partly reflect the nature of this vibrant and quickly evolving new field.

GOALS

The ultimate goal of bioinformatics is to better understand a living cell and how it functions at the molecular level. By analyzing raw molecular sequence and structural data, bioinformatics research can generate new insights and provide a "global" perspective of the cell. The reason that the functions of a cell can be better understood by analyzing sequence data is ultimately because the flow of genetic information is dictated by the "central dogma" of biology in which DNA is transcribed to RNA, which is translated to proteins. Cellular functions are mainly performed by proteins whose capabilities are ultimately determined by their sequences. Therefore, solving functional problems using sequence and sometimes structural approaches has proved to be a fruitful endeavor.

SCOPE

Bioinformatics consists of two subfields: the development of computational tools and databases and the application of these tools and databases in generating biological knowledge to better understand living systems. These two subfields are complementary to each other. The tool development includes writing software for sequence, structural, and functional analysis, as well as the construction and curating of biological databases. These tools are used in three areas of genomic and molecular biological research: molecular sequence analysis, molecular structural analysis, and molecular functional analysis. The analyses of biological data often generate new problems and challenges that in turn spur the development of new and better computational tools.

The areas of sequence analysis include sequence alignment, sequence database searching, motif and pattern discovery, gene and promoter finding, reconstruction of evolutionary relationships, and genome assembly and comparison. Structural analyses include protein and nucleic acid structure analysis, comparison, classification, and prediction. The functional analyses include gene expression profiling, protein–protein interaction prediction, protein subcellular localization prediction, metabolic pathway reconstruction, and simulation (Fig. 1.1).

The three aspects of bioinformatics analysis are not isolated but often interact to produce integrated results (see Fig. 1.1). For example, protein structure prediction depends on sequence alignment data; clustering of gene expression profiles requires the use of phylogenetic tree construction methods derived in sequence analysis. Sequence-based promoter prediction is related to functional analysis of

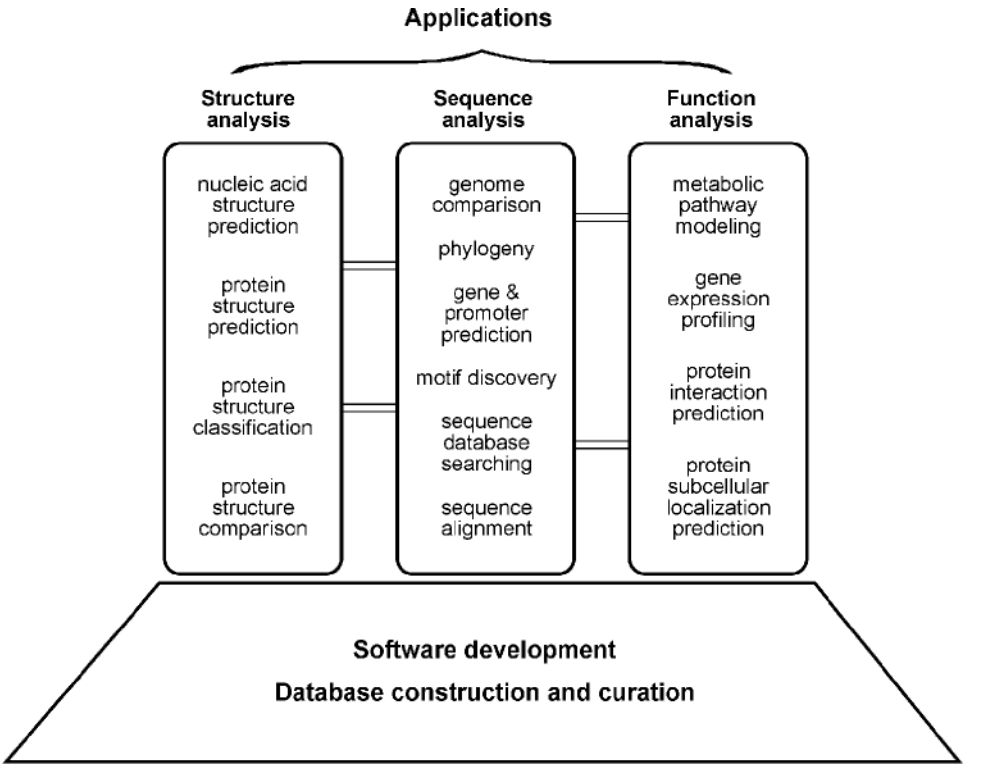

Figure 1.1: Overview of various subfields of bioinformatics. Biocomputing tool development is at the foundation of all bioinformatics analysis. The applications of the tools fall into three areas: sequence analysis, structure analysis, and function analysis. There are intrinsic connections between different areas of analyses represented by bars between the boxes.

coexpressed genes. Gene annotation involves a number of activities, which include distinction between coding and noncoding sequences, identification of translated protein sequences, and determination of the gene's evolutionary relationship with other known genes; prediction of its cellular functions employs tools from all three groups of the analyses.

APPLICATIONS

Bioinformatics has not only become essential for basic genomic and molecular biology research, but is having a major impact on many areas of biotechnology and biomedical sciences. It has applications, for example, in knowledge-based drug design, forensic DNA analysis, and agricultural biotechnology. Computational studies of protein–ligand interactions provide a rational basis for the rapid identification of novel leads for synthetic drugs. Knowledge of the three-dimensional structures of proteins allows molecules to be designed that are capable of binding to the receptor site of a target protein with great affinity and specificity. This informatics-based approach

significantly reduces the time and cost necessary to develop drugs with higher potency, fewer side effects, and less toxicity than using the traditional trial-and-error approach. In forensics, results from molecular phylogenetic analysis have been accepted as evidence in criminal courts. Some sophisticated Bayesian statistics and likelihood-based methods for analysis of DNA have been applied in the analysis of forensic identity. It is worth mentioning that genomics and bioinformtics are now poised to revolutionize our healthcare system by developing personalized and customized medicine. The high speed genomic sequencing coupled with sophisticated informatics technology will allow a doctor in a clinic to quickly sequence a patient's genome and easily detect potential harmful mutations and to engage in early diagnosis and effective treatment of diseases. Bioinformatics tools are being used in agriculture as well. Plant genome databases and gene expression profile analyses have played an important role in the development of new crop varieties that have higher productivity and more resistance to disease.

LIMITATIONS

Having recognized the power of bioinformatics, it is also important to realize its limitations and avoid over-reliance on and over-expectation of bioinformatics output. In fact, bioinformatics has a number of inherent limitations. In many ways, the role of bioinformatics in genomics and molecular biology research can be likened to the role of intelligence gathering in battlefields. Intelligence is clearly very important in leading to victory in a battlefield. Fighting a battle without intelligence is inefficient and dangerous. Having superior information and correct intelligence helps to identify the enemy's weaknesses and reveal the enemy's strategy and intentions. The gathered information can then be used in directing the forces to engage the enemy and win the battle. However, completely relying on intelligence can also be dangerous if the intelligence is of limited accuracy. Overreliance on poor-quality intelligence can yield costly mistakes if not complete failures.

It is no stretch in analogy that fighting diseases or other biological problems using bioinformatics is like fighting battles with intelligence. Bioinformatics and experimental biology are independent, but complementary, activities. Bioinformatics depends on experimental science to produce raw data for analysis. It, in turn, provides useful interpretation of experimental data and important leads for further experimental research. Bioinformatics predictions are not formal proofs of any concepts. They do not replace the traditional experimental research methods of actually testing hypotheses. In addition, the quality of bioinformatics predictions depends on the quality of data and the sophistication of the algorithms being used. Sequence data from high throughput analysis often contain errors. If the sequences are wrong or annotations incorrect, the results from the downstream analysis are misleading as well. That is why it is so important to maintain a realistic perspective of the role of bioinformatics.

Bioinformatics is by no means a mature field. Most algorithms lack the capability and sophistication to truly reflect reality. They often make incorrect predictions that make no sense when placed in a biological context. Errors in sequence alignment, for example, can affect the outcome of structural or phylogenetic analysis. The outcome of computation also depends on the computing power available. Many accurate but exhaustive algorithms cannot be used because of the slow rate of computation. Instead, less accurate but faster algorithms have to be used. This is a necessary trade-off between accuracy and computational feasibility. Therefore, it is important to keep in mind the potential for errors produced by bioinformatics programs. Caution should always be exercised when interpreting prediction results. It is a good practice to use multiple programs, if they are available, and perform multiple evaluations. A more accurate prediction can often be obtained if one draws a consensus by comparing results from different algorithms.

NEW THEMES

Despite the pitfalls, there is no doubt that bioinformatics is a field that holds great potential for revolutionizing biological research in the coming decades. Currently, the field is undergoing major expansion. In addition to providing more reliable and more rigorous computational tools for sequence, structural, and functional analysis, the major challenge for future bioinformatics development is to develop tools for elucidation of the functions and interactions of all gene products in a cell. This presents a tremendous challenge because it requires integration of disparate fields of biological knowledge and a variety of complex mathematical and statistical tools. To gain a deeper understanding of cellular functions, mathematical models are needed to simulate a wide variety of intracellular reactions and interactions at the whole cell level. This molecular simulation of all the cellular processes is termed *systems biology*. Achieving this goal will represent a major leap toward fully understanding a living system. That is why the system-level simulation and integration are considered the future of bioinformatics. Modeling such complex networks and making predictions about their behavior present tremendous challenges and opportunities for bioinformaticians. The ultimate goal of this endeavor is to transform biology from a qualitative science to a quantitative and predictive science. This is truly an exciting time for bioinformatics.

FURTHER READING

Attwood, T. K., and Miller, C. J. 2002. Progress in bioinformatics and the importance of being earnest. *Biotechnol. Annu. Rev*. 8:1–54.

Golding, G. B. 2003. DNA and the revolution of molecular evolution, computational biology, and bioinformatics. *Genome* 46:930–5.

Goodman, N. 2002. Biological data becomes computer literature: New advances in bioinformatics. *Curr. Opin. Biotechnol.* 13:68–71.

Hagen. J. B. 2000. The origin of bioinformatics. *Nat. Rev. Genetics* 1:231–6.

Kanehisa, M., and Bork, P. 2003. Bioinformatics in the post-sequence era. *Nat. Genet*. 33 Suppl:305–10.

Kim, J. H. 2002. Bioinformatics and genomic medicine. *Genet. Med.* 4 Suppl:62S–5S.

Luscombe, N. M., Greenbaum, D., and Gerstein, M. 2001. What is bioinformatics? A proposed definition and overview of the field. *Methods Inf. Med*. 40:346–58.

Ouzounis, C. A., and Valencia, A. 2003. Early bioinformatics: The birth of a discipline – A personal view. *Bioinformatics* 19:2176–90.

CHAPTER TWO

Introduction to Biological Databases

One of the hallmarks of modern genomic research is the generation of enormous amounts of raw sequence data. As the volume of genomic data grows, sophisticated computational methodologies are required to manage the data deluge. Thus, the very first challenge in the genomics era is to store and handle the staggering volume of information through the establishment and use of computer databases. The development of databases to handle the vast amount of molecular biological data is thus a fundamental task of bioinformatics. This chapter introduces some basic concepts related to databases, in particular, the types, designs, and architectures of biological databases. Emphasis is on retrieving data from the main biological databases such as GenBank.

WHAT IS A DATABASE?

A *database* is a computerized archive used to store and organize data in such a way that information can be retrieved easily via a variety of search criteria. Databases are composed of computer hardware and software for data management. The chief objective of the development of a database is to organize data in a set of structured records to enable easy retrieval of information. Each record, also called an *entry*, should contain a number of fields that hold the actual data items, for example, fields for names, phone numbers, addresses, dates. To retrieve a particular record from the database, a user can specify a particular piece of information, called *value*, to be found in a particular field and expect the computer to retrieve the whole data record. This process is called *making a query*.

Although data retrieval is the main purpose of all databases, biological databases often have a higher level of requirement, known as *knowledge discovery*, which refers to the identification of connections between pieces of information that were not known when the information was first entered. For example, databases containing raw sequence information can perform extra computational tasks to identify sequence homology or conserved motifs. These features facilitate the discovery of new biological insights from raw data.

TYPES OF DATABASES

Originally, databases all used a flat file format, which is a long text file that contains many entries separated by a *delimiter*, a special character such as a vertical bar (|). Within each entry are a number of fields separated by tabs or commas. Except for the

raw values in each field, the entire text file does not contain any hidden instructions for computers to search for specific information or to create reports based on certain fields from each record. The text file can be considered a single table. Thus, to search a flat file for a particular piece of information, a computer has to read through the entire file, an obviously inefficient process. This is manageable for a small database, but as database size increases or data types become more complex, this database style can become very difficult for information retrieval. Indeed, searches through such files often cause crashes of the entire computer system because of the memory-intensive nature of the operation.

To facilitate the access and retrieval of data, sophisticated computer software programs for organizing, searching, and accessing data have been developed. They are called *database management systems*. These systems contain not only raw data records but also operational instructions to help identify hidden connections among data records. The purpose of establishing a data structure is for easy execution of the searches and to combine different records to form final search reports. Depending on the types of data structures,these database management systems can be classified into two types: *relational database management systems* and *object-oriented database management systems*. Consequently, databases employing these management systems are known as *relational databases* or *object-oriented databases*, respectively.

Relational Databases

Instead of using a single table as in a flat file database, relational databases use a set of tables to organize data. Each table, also called a *relation*, is made up of columns and rows. Columns represent individual fields. Rows represent values in the fields of records. The columns in a table are indexed according to a common feature called an *attribute*, so they can be cross-referenced in other tables. To execute a query in a relational database, the system selects linked data items from different tables and combines the information into one report. Therefore, specific information can be found more quickly from a relational database than from a flat file database.

Relational databases can be created using a special programming language called *structured query language* (SQL). The creation of this type of databases can take a great deal of planning during the design phase. After creation of the original database, a new data category can be easily added without requiring all existing tables to be modified. The subsequent database searching and data gathering for reports are relatively straightforward.

Here is a simple example of student course information expressed in a flat file which contains records of five students from four different states, each taking a different course (Fig. 2.1). Each data record, separated by a vertical bar, contains four fields describing the name, state, course number and title. A relational database is also created to store the same information, in which the data are structured as a number of tables. Figure 2.1 shows how the relational database works. In each table, data that fit a particular criterion are grouped together. Different tables can be linked by common data categories, which facilitate finding of specific information.

Flat File

Name, States, Course number, Course name|John Smith, Texas, Biol 689, Bioinformatics|Jane Doe, Kansas, Bich 441, Biochemistry|William Brown, Illinois, Chem 289, Organic Chemistry|Jennifer Taylor, New York, Hort 201, Horticulture|Howard Douglas, Texas, Math 172, Calculus

Table A

Student #	Name	State
1	John Smith	Texas
2	Jane Doe	Kansas
3	William Brown	Illinois
4	Jennifer Taylor	New York
5	Howard Douglas	Texas

Table B

Student #	Course #
1	Biol 689
2	Bich 441
3	Chem 289
4	Hort 201
5	Math 172

Table C

Course #	Course name
Biol 689	Bioinformatics
Bich 441	Biochemistry
Chem 289	Organic chemistry
Hort 201	Horticulture
Math 172	Calculus

Figure 2.1: Example of constructing a relational database for five students' course information originally expressed in a flat file. By creating three different tables linked by common fields, data can be easily accessed and reassembled.

For example, if one is to ask the question, which courses are students from Texas taking? The database will first find the field for "State" in Table A and look up for Texas. This returns students 1 and 5. The student numbers are colisted in Table B, in which students 1 and 5 correspond to Biol 689 and Math 172, respectively. The course names listed by course numbers are found in Table C. By going to Table C, exact course names corresponding to the course numbers can be retrieved. A final report is then given showing that the Texans are taking the courses Bioinformatics and Calculus. However, executing the same query through the flat file requires the computer to read through the entire text file word by word and to store the information in a temporay memory space and later mark up the data records containing the word *Texas*. This is easily accomplishable for a small database. To perform queries in a large database using flat files obviously becomes an onerous task for the computer system.

Object-Oriented Databases

One of the problems with relational databases is that the tables used do not describe complex hierarchical relationships between data items. To overcome the problem, object-oriented databases have been developed that store data as objects. In an object-oriented programming language, an object can be considered as a unit that combines data and mathematical routines that act on the data. The database is structured such that the objects are linked by a set of pointers defining predetermined relationships between the objects. Searching the database involves navigating through the objects with the aid of the pointers linking different objects. Programming languages like C++ are used to create object-oriented databases.

The object-oriented database system is more flexible; data can be structured based on hierarchical relationships. By doing so, programming tasks can be simplified for data that are known to have complex relationships, such as multimedia data. However,

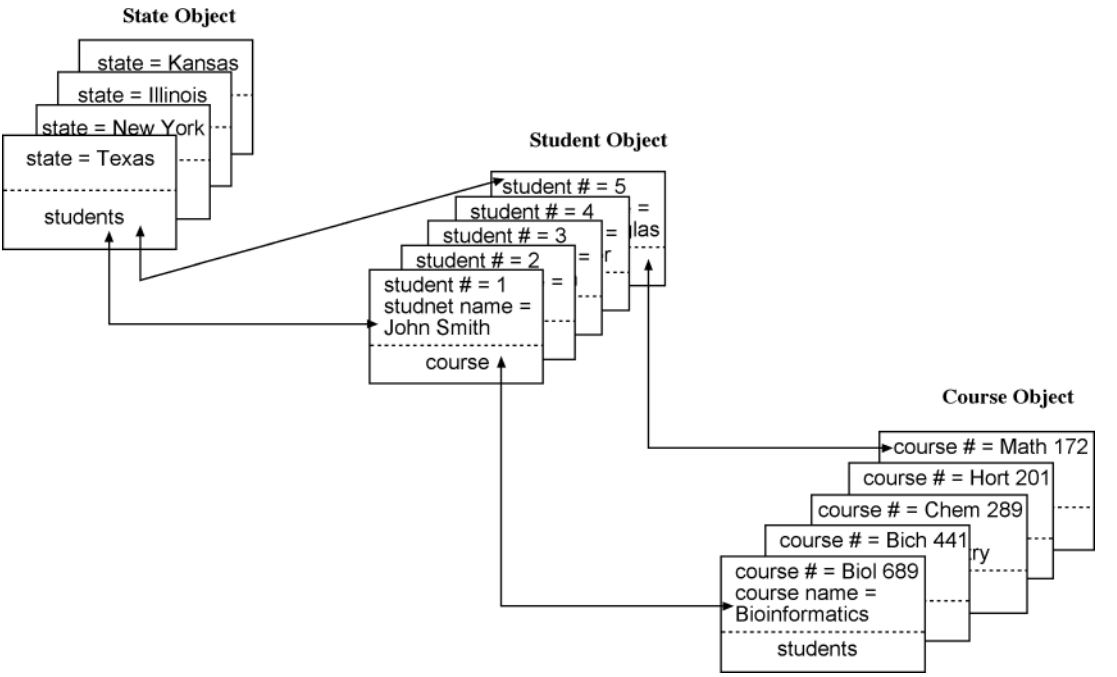

Figure 2.2: Example of construction and query of an object-oriented database using the same student information as shown in Figure 2.1. Three objects are constructed and are linked by pointers shown as arrows. Finding specific information relies on navigating through the objects by way of pointers. For simplicity, some of the pointers are omitted.

this type of database system lacks the rigorous mathematical foundation of the relational databases. There is also a risk that some of the relationships between objects may be misrepresented. Some current databases have therefore incorporated features of both types of database programming, creating the *object–relational database management system*.

The above students' course information (Fig. 2.1) can be used to construct an object-oriented database. Three different objects can be designed: student object, course object, and state object. Their interrelations are indicated by lines with arrows (Fig. 2.2). To answer the same question – which courses are students from Texas taking – one simply needs to start from Texas in the state object, which has pointers that lead to students 1 and 5 in the student object. Further pointers in the student object point to the course each of the two students is taking. Therefore, a simple navigation through the linked objects provides a final report.

BIOLOGICAL DATABASES

Current biological databases use all three types of database structures: flat files, relational, and object oriented. Despite the obvious drawbacks of using flat files in database management, many biological databases still use this format. The justification for this is that this system involves minimum amount of database design and the search output can be easily understood by working biologists.

Based on their contents, biological databases can be roughly divided into three categories: primary databases, secondary databases, and specialized databases. *Primary databases* contain original biological data. They are archives of raw sequence or structural data submitted by the scientific community. GenBank and Protein Data Bank (PDB) are examples of primary databases. *Secondary databases* contain computationally processed or manually curated information, based on original information from primary databases. Translated protein sequence databases containing functional annotation belong to this category. Examples are SWISS-Prot and Protein Information Resources (PIR) (successor of Margaret Dayhoff's Atlas of Protein Sequence and Structure [see Chapter 1]). Specialized databases are those that cater to a particular research interest. For example, Flybase, HIV sequence database, and Ribosomal Database Project are databases that specialize in a particular organism or a particular type of data. A list of some frequently used databases is provided in Table 2.1.

Primary Databases

There are three major public sequence databases that store raw nucleic acid sequence data produced and submitted by researchers worldwide: GenBank, the European Molecular Biology Laboratory (EMBL) database and the DNA Data Bank of Japan (DDBJ), which are all freely available on the Internet. Most of the data in the databases are contributed directly by authors with a minimal level of annotation. A small number of sequences, especially those published in the 1980s, were entered manually from published literature by database management staff.

Presently, sequence submission to either GenBank, EMBL, or DDBJ is a precondition for publication in most scientific journals to ensure the fundamental molecular data to be made freely available. These three public databases closely collaborate and exchange new data daily. They together constitute the International Nucleotide Sequence Database Collaboration. This means that by connecting to any one of the three databases, one should have access to the same nucleotide sequence data. Although the three databases all contain the same sets of raw data, each of the individual databases has a slightly different kind of format to represent the data.

Fortunately, for the three-dimensional structures of biological macromolecules, there is only one centralized database, the PDB. This database archives atomic coordinates of macromolecules (both proteins and nucleic acids) determined by x-ray crystallography and NMR. It uses a flat file format to represent protein name, authors, experimental details, secondary structure, cofactors, and atomic coordinates. The web interface of PDB also provides viewing tools for simple image manipulation. More details of this database and its format are provided in Chapter 12.

Secondary Databases

Sequence annotation information in the primary database is often minimal. To turn the raw sequence information into more sophisticated biological knowledge, much postprocessing of the sequence information is needed. This begs the need for

TABLE 2.1. Major Biological Databases Available Via the World Wide Web

Databases and Retrieval Systems	Brief Summary of Content	URL
AceDB	Genome database for *Caenorhabditis elegans*	www.acedb.org
DDBJ	Primary nucleotide sequence database in Japan	www.ddbj.nig.ac.jp
EMBL	Primary nucleotide sequence database in Europe	www.ebi.ac.uk/embl/index.html
Entrez	NCBI portal for a variety of biological databases	www.ncbi.nlm.nih.gov/gquery/gquery.fcgi
ExPASY	Proteomics database	http://us.expasy.org/
FlyBase	A database of the *Drosophila* genome	http://flybase.bio.indiana.edu/
FSSP	Protein secondary structures	www.bioinfo.biocenter.helsinki.fi:8080/dali/index.html
GenBank	Primary nucleotide sequence database in NCBI	www.ncbi.nlm.nih.gov/Genbank
HIV databases	HIV sequence data and related immunologic information	www.hiv.lanl.gov/content/index
Microarray gene expression database	DNA microarray data and analysis tools	www.ebi.ac.uk/microarray
OMIM	Genetic information of human diseases	www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM
PIR	Annotated protein sequences	http://pir.georgetown.edu/pirwww/pirhome3.shtml
PubMed	Biomedical literature information	www.ncbi.nlm.nih.gov/PubMed
Ribosomal database project	Ribosomal RNA sequences and phylogenetic trees derived from the sequences	http://rdp.cme.msu.edu/html
SRS	General sequence retrieval system	http://srs6.ebi.ac.uk
SWISS-Prot	Curated protein sequence database	www.ebi.ac.uk/swissprot/access.html
TAIR	Arabidopsis information database	www.arabidopsis.org

secondary databases, which contain computationally processed sequence information derived from the primary databases. The amount of computational processing work varies greatly among the secondary databases; some are simple archives of translated sequence data from identified open reading frames in DNA, whereas others provide additional annotation and information related to higher levels of information regarding structure and functions.

A prominent example of secondary databases is SWISS-PROT, which provides detailed sequence annotation that includes structure, function, and protein family assignment. The sequence data are mainly derived from TrEMBL, a database of

translated nucleic acid sequences stored in the EMBL database. The annotation of each entry is carefully curated by human experts and thus is of good quality. The protein annotation includes function, domain structure, catalytic sites, cofactor binding, posttranslational modification, metabolic pathway information, disease association, and similarity with other sequences. Much of this information is obtained from scientific literature and entered by database curators. The annotation provides significant added value to each original sequence record. The data record also provides cross-referencing links to other online resources of interest. Other features such as very low redundancy and high level of integration with other primary and secondary databases make SWISS-PROT very popular among biologists.

A recent effort to combine SWISS-PROT, TrEMBL, and PIR led to the creation of the UniProt database, which has larger coverage than any one of the three databases while at the same time maintaining the original SWISS-PROT feature of low redundancy, cross-references, and a high quality of annotation.

There are also secondary databases that relate to protein family classification according to functions or structures. The Pfam and Blocks databases (to be described in Chapter 7) contain aligned protein sequence information as well as derived motifs and patterns, which can be used for classification of protein families and inference of protein functions. The DALI database (to be described in Chapter 13) is a protein secondary structure database that is vital for protein structure classification and threading analysis (to be described in Chapter 15) to identify distant evolutionary relationships among proteins.

Specialized Databases

Specialized databases normally serve a specific research community or focus on a particular organism. The content of these databases may be sequences or other types of information. The sequences in these databases may overlap with a primary database, but may also have new data submitted directly by authors. Because they are often curated by experts in the field, they may have unique organizations and additional annotations associated with the sequences. Many genome databases that are taxonomic specific fall within this category. Examples include Flybase, WormBase, AceDB, and TAIR (Table 2.1). In addition, there are also specialized databases that contain original data derived from functional analysis. For example, GenBank EST database and Microarray Gene Expression Database at the European Bioinformatics Institute (EBI) are some of the gene expression databases available.

Interconnection between Biological Databases

As mentioned, primary databases are central repositories and distributors of raw sequence and structure information. They support nearly all other types of biological databases in a way akin to the Associated Press providing news feeds to local news media, which then tailor the news to suit their own particular needs. Therefore, in the biological community, there is a frequent need for the secondary and specialized

databases to connect to the primary databases and to keep uploading sequence information. In addition, a user often needs to get information from both primary and secondary databases to complete a task because the information in a single database is often insufficient. Instead of letting users visiting multiple databases, it is convenient for entries in a database to be cross-referenced and linked to related entries in other databases that contain additional information. All these create a demand for linking different databases.

The main barrier to linking different biological databases is format incompatibility current biological databases utilize all three types of database structures – flat files, relational, and object oriented. The heterogeneous database structures limit communication between databases. One solution to networking the databases is to use a specification language called Common Object Request Broker Architecture (COBRA), which allows database programs at different locations to communicate in a network through an "interface broker" without having to understand each other's database structure. It works in a way similar to HyperText Markup Language (HTML) for web pages, labeling database entries using a set of common tags.

A similar protocol called eXtensible Markup Language (XML) also helps in bridging databases. In this format, each biological record is broken down into small, basic components that are labeled with a hierarchical nesting of tags. This database structure significantly improves the distribution and exchange of complex sequence annotations between databases. Recently, a specialized protocol for bioinformatics data exchange has been developed. It is the distributed annotation system, which allows one computer to contact multiple servers and retrieve dispersed sequence annotation information related to a particular sequence and integrate the results into a single combined report.

PITFALLS OF BIOLOGICAL DATABASES

One of the problems associated with biological databases is overreliance on sequence information and related annotations, without understanding the reliability of the information. What is often ignored is the fact that there are many errors in sequence databases. There are also high levels of redundancy in the primary sequence databases. Annotations of genes can also occasionally be false or incomplete. All these types of errors can be passed on to other databases, causing propagation of errors.

Most errors in nucleotide sequences are caused by sequencing errors. Some of these errors cause frameshifts that make whole gene identification difficult or protein translation impossible. Sometimes, gene sequences are contaminated with sequences from cloning vectors. Generally speaking, errors are more common for sequences produced before the 1990s; sequence quality has been greatly improved since. Therefore, exceptional care should be taken when dealing with more dated sequences.

Redundancy is another major problem affecting primary databases. There is tremendous duplication of information in the databases, for various reasons. The

causes of redundancy include repeated submission of identical or overlapping sequences by the same or different authors, revision of annotations, dumping of expressed sequence tags (EST) data (see Chapter 18), and poor database management that fails to detect the redundancy. This makes some primary databases excessively large and unwieldy for information retrieval.

Steps have been taken to reduce the redundancy. The National Center for Biotechnology Information (NCBI) has now created a *nonredundant* database, called RefSeq, in which identical sequences from the same organism and associated sequence fragments are merged into a single entry. Proteins sequences derived from the same DNA sequences are explicitly linked as related entries. Sequence variants from the same organism with very minor differences, which may well be caused by sequencing errors, are treated as distinctly related entries. This carefully curated database can be considered a secondary database.

As mentioned, the SWISS-PROT database also has minimal redundancy for protein sequences compared to most other databases. Another way to address the redundancy problem is to create sequence-cluster databases such as UniGene (see Chapter 18) that coalesce EST sequences that are derived from the same gene.

The other common problem is erroneous annotations. Often, the same gene sequence is found under different names resulting in multiple entries and confusion about the data. Or conversely, unrelated genes bearing the same name are found in the databases. To alleviate the problem of naming genes, reannotation of genes and proteins using a set of common, controlled vocabulary to describe a gene or protein is necessary. The goal is to provide a consistent and unambiguous naming system for all genes and proteins. A prominent example of such systems is *Gene Ontology* (see Chapter 17).

Some of the inconsistencies in annotation could be caused by genuine disagreement between researchers in the field; others may result from imprudent assignment of protein functions by sequence submitters. There are also some errors that are simply caused by omissions or mistakes in typing. Errors in annotation can be particularly damaging because the large majority of new sequences are assigned functions based on similarity with sequences in the databases that are already annotated. Therefore, a wrong annotation can be easily transferred to all similar genes in the entire database. It is possible that some of these errors can be corrected at the informatics level by studying the protein domains and families. However, others eventually have to be corrected using experimental work.

INFORMATION RETRIEVAL FROM BIOLOGICAL DATABASES

As mentioned, a major goal in developing databases is to provide efficient and user-friendly access to the data stored. There are a number of retrieval systems for biological data. The most popular retrieval systems for biological databases are Entrez and Sequence Retrieval Systems (SRS) that provide access to multiple databases for retrieval of integrated search results.

To perform complex queries in a database often requires the use of Boolean operators. This is to join a series of keywords using logical terms such as AND, OR, and NOT to indicate relationships between the keywords used in a search. *AND* means that the search result must contain both words; *OR* means to search for results containing either word or both; *NOT* excludes results containing either one of the words. In addition, one can use parentheses () to define a concept if multiple words and relationships are involved, so that the computer knows which part of the search to execute first. Items contained within parentheses are executed first. Quotes can be used to specify a phrase. Most search engines of public biological databases use some form of this Boolean logic.

Entrez

The NCBI developed and maintains Entrez, a biological database retrieval system. It is a gateway that allows text-based searches for a wide variety of data, including annotated genetic sequence information, structural information, as well as citations and abstracts, full papers, and taxonomic data. The key feature of Entrez is its ability to integrate information, which comes from cross-referencing between NCBI databases based on preexisting and logical relationships between individual entries. This is highly convenient: users do not have to visit multiple databases located in disparate places. For example, in a nucleotide sequence page, one may find cross-referencing links to the translated protein sequence, genome mapping data, or to the related PubMed literature information, and to protein structures if available.

Effective use of Entrez requires an understanding of the main features of the search engine. There are several options common to all NCBI databases that help to narrow the search. One option is "Limits," which helps to restrict the search to a subset of a particular database. It can also be set to restrict a search to a particular database (e.g., the field for author or publication date) or a particular type of data (e.g., chloroplast DNA/RNA). Another option is "Preview/Index," which connects different searches with the Boolean operators and uses a string of logically connected keywords to perform a new search. The search can also be limited to a particular search field (e.g., gene name or accession number). The "History" option provides a record of the previous searches so that the user can review, revise, or combine the results of earlier searches. There is also a "Clipboard" that stores search results for later viewing for a limited time. To store information in the Clipboard, the "Send to Clipboard" function should be used.

One of the databases accessible from Entrez is a biomedical literature database known as PubMed, which contains abstracts and in some cases the full text articles from nearly 4,000 journals. An important feature of PubMed is the retrieval of information based on medical subject headings (MeSH) terms. The MeSH system consists of a collection of more than 20,000 controlled and standardized vocabulary terms used for indexing articles. In other words, it is a thesaurus that helps convert search keywords into standardized terms to describe a concept. By doing so, it allows "smart" searches in which a group of accepted synonyms are employed so that the user not only gets

TABLE 2.2. Several Selected PubMed Tags and Their Brief Descriptions

Tag	Name	Description
AB	Abstract	Abstract
AD	Affiliation	Institutional affiliation and address of the first author and grant numbers
AID	Article identifier	Article ID values may include the PII (controlled publisher identifier) or doi (digital object identifier)
AU	Author	Authors
DP	Publication date	The date the article was published
JID	Journal ID	Unique journal ID in the National Library of Medicine's catalog of books, journals, and audiovisuals
LA	Language	The language in which the article was published
PL	Place of publication	Journal's country of publication
PT	Publication type	The type of material the article represents
RN	EC/RN number	Number assigned by the Enzyme Commission to designate a particular enzyme or by the Chemical Abstracts Service for Registry Numbers
SO	Source	Composite field containing bibliographic information
TA	Journal title abbreviation	Standard journal title abbreviation
TI	Title	The title of the article
VI	Volume	Journal volume

Source: www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhelp.html.

exact matches, but also related matches on the same topic that otherwise might have been missed. Another way to broaden the retrieval is by using the "Related Articles" option. PubMed uses a word weight algorithm to identify related articles with similar words in the titles, abstracts, and MeSH. By using this feature, articles on the same topic that were missed in the original search can be retrieved.

For a complex search, a user can use the Boolean operators or a combination of Limits and Preview/Index features to conduct complex searches. Alternatively, field tags can be used to improve the efficiency of obtaining the search results. The tags are identifiers for each field and are placed in brackets. For example, [AU] limits the search for author name, and [JID] for journal name. PubMed uses a list of tags for literature searches. The search terms can be specified by the tags which are joined by Boolean operators. Some frequently used PubMed field tags are given in Table 2.2.

Another unique database accessible from Entrez is Online Mendelian Inheritance in Man (OMIM), which is a non-sequence-based database of human disease genes and human genetic disorders. Each entry in OMIM contains summary information about a particular disease as well as genes related to the disease. The text contains numerous hyperlinks to literature citations, primary sequence records, as well as chromosome loci of the disease genes. The database can serve as an excellent starting point to study genes related to a disease.

NCBI also maintains a taxonomy database that contains the names and taxonomic positions of over 100,000 organisms with at least one nucleotide or protein sequence

represented in the GenBank database. The taxonomy database has a hierarchical classification scheme. The root level is Archaea, Eubacteria, and Eukaryota. The database allows the taxonomic tree for a particular organism to be displayed. The tree is based on molecular phylogenetic data, namely, the small ribosomal RNA data.

GenBank

GenBank is the most complete collection of annotated nucleic acid sequence data for almost every organism. The content includes genomic DNA, mRNA, cDNA, ESTs, high throughput raw sequence data, and sequence polymorphisms. There is also a GenPept database for protein sequences, the majority of which are conceptual translations from DNA sequences, although a small number of the amino acid sequences are derived using peptide sequencing techniques.

There are two ways to search for sequences in GenBank. One is using text-based keywords similar to a PubMed search. The other is using molecular sequences to search by sequence similarity using BLAST (to be described in Chapter 5).

GenBank Sequence Format

To search GenBank effectively using the text-based method requires an understanding of the GenBank sequence format. GenBank is a relational database. However, the search output for sequence files is produced as flat files for easy reading. The resulting flat files contain three sections – Header, Features, and Sequence entry (Fig. 2.3). There are many fields in the Header and Features sections. Each field has an unique identifier for easy indexing by computer software. Understanding the structure of the GenBank files helps in designing effective search strategies.

The Header section describes the origin of the sequence, identification of the organism, and unique identifiers associated with the record. The top line of the Header section is the Locus, which contains a unique database identifier for a sequence location in the database (not a chromosome locus). The identifier is followed by sequence length and molecule type (e.g., DNA or RNA). This is followed by a three-letter code for GenBank divisions. There are 17 divisions in total, which were set up simply based on convenience of data storage without necessarily having rigorous scientific basis; for example, PLN for plant, fungal, and algal sequences; PRI for primate sequences; MAM for nonprimate mammalian sequences; BCT for bacterial sequences; and EST for EST sequences. Next to the division is the date when the record was made public (which is different from the date when the data were submitted).

The following line, "DEFINITION," provides the summary information for the sequence record including the name of the sequence, the name and taxonomy of the source organism if known, and whether the sequence is complete or partial. This is followed by an accession number for the sequence, which is a unique number assigned to a piece of DNA when it was first submitted to GenBank and is permanently associated with that sequence. This is the number that should be cited in publications. It has two different formats: two letters with five digits or one letter with six digits. For a nucleotide sequence that has been translated into a protein sequence,

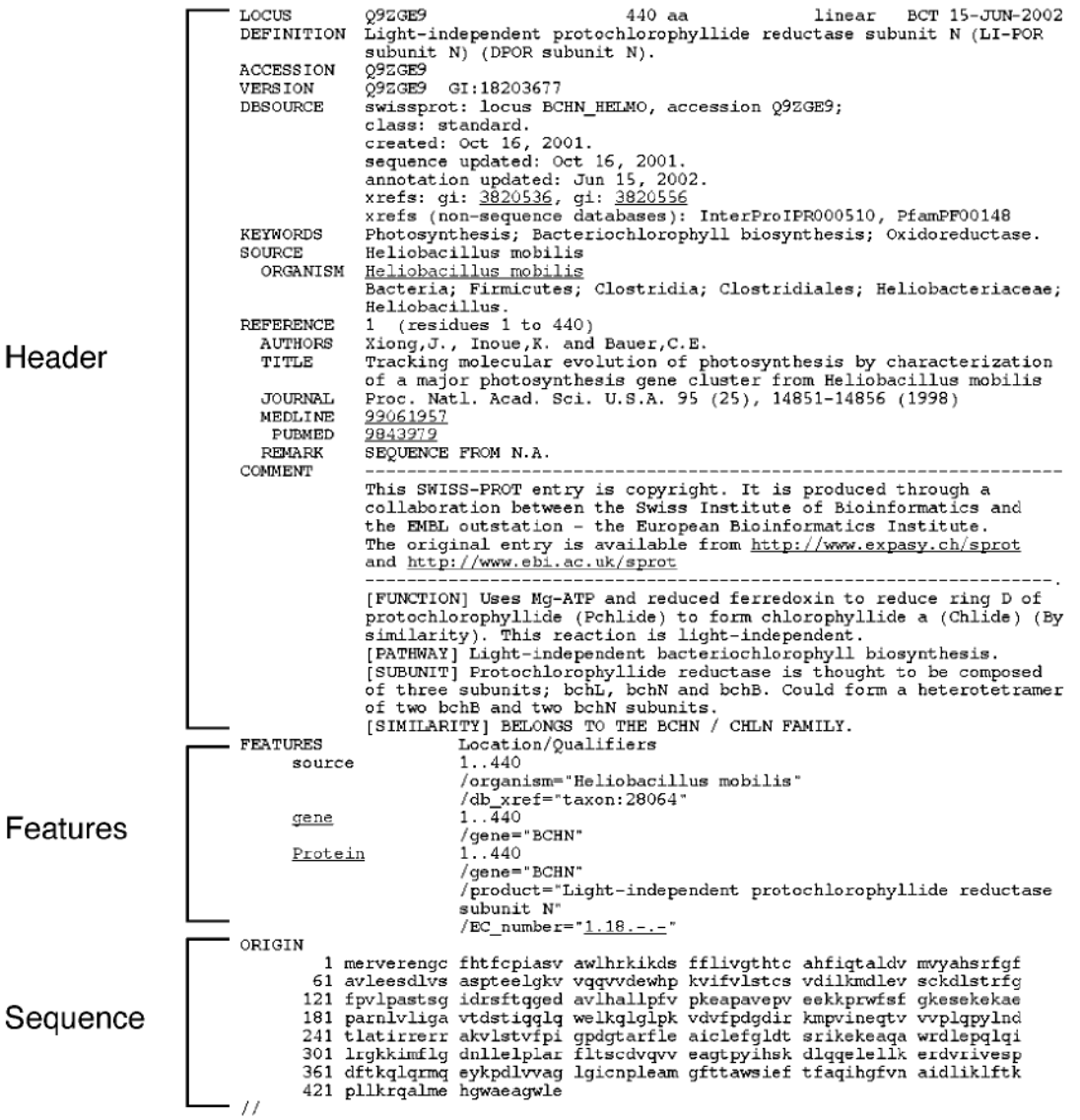

Figure 2.3: NCBI GenBank/GenPept format showing the three major components of a sequence file.

a new accession number is given in the form of a string of alphanumeric characters. In addition to the accession number, there is also a version number and a gene index (gi) number. The purpose of these numbers is to identify the current version of the sequence. If the sequence annotation is revised at a later date, the accession number remains the same, but the version number is incremented as is the gi number. A translated protein sequence also has a different gi number from the DNA sequence it is derived from.

The next line in the Header section is the "ORGANISM" field, which includes the source of the organism with the scientific name of the species and sometimes the

tissue type. Along with the scientific name is the information of taxonomic classification of the organism. Different levels of the classification are hyperlinked to the NCBI taxonomy database with more detailed descriptions. This is followed by the "REFERENCE" field, which provides the publication citation related to the sequence entry. The REFERENCE part includes author and title information of the published work (or tentative title for unpublished work). The "JOURNAL" field includes the citation information as well as the date of sequence submission. The citation is often hyperlinked to the PubMed record for access to the original literature information. The last part of the Header is the contact information of the sequence submitter.

The "Features" section includes annotation information about the gene and gene product, as well as regions of biological significance reported in the sequence, with identifiers and qualifiers. The "Source" field provides the length of the sequence, the scientific name of the organism, and the taxonomy identification number. Some optional information includes the clone source, the tissue type and the cell line. The "gene" field is the information about the nucleotide coding sequence and its name. For DNA entries, there is a "CDS" field, which is information about the boundaries of the sequence that can be translated into amino acids. For eukaryotic DNA, this field also contains information of the locations of exons and translated protein sequences is entered.

The third section of the flat file is the sequence itself starting with the label "ORIGIN." The format of the sequence display can be changed by choosing options at a Display pull-down menu at the upper left corner. For DNA entries, there is a BASE COUNT report that includes the numbers of A, G, C, and T in the sequence. This section, for both DNA or protein sequences, ends with two forward slashes (the "//" symbol).

In retrieving DNA or protein sequences from GenBank, the search can be limited to different fields of annotation such as "organism," "accession number," "authors," and "publication date." One can use a combination of the "Limits" and "Preview/Index" options as described. Alternatively, a number of search qualifiers can be used, each defining one of the fields in a GenBank file. The qualifiers are similar to but not the same as the field tags in PubMed. For example, in GenBank, [GENE] represents field for gene name, [AUTH] for author name, and [ORGN] for organism name. Frequently used GenBank qualifiers, which have to be in uppercase and in brackets, are listed in Table 2.3.

Alternative Sequence Formats

FASTA. In addition to the GenBank format, there are many other sequence formats. FASTA is one of the simplest and the most popular sequence formats because it contains plain sequence information that is readable by many bioinformatics analysis programs. It has a single definition line that begins with a right angle bracket (>) followed by a sequence name (Fig. 2.4). Sometimes, extra information such as gi number or comments can be given, which are separated from the sequence name by a "|" symbol. The extra information is considered optional and is ignored by

TABLE 2.3. Search Field Qualifiers for GenBank

Qualifier	Field Name	Definition
[ACCN]	Accession	Contains the unique accession number of the sequence or record, assigned to the nucleotide, protein, structure, or genome record.
[ALL]	All fields	Contains all terms from all searchable database fields in the database.
[AUTH]	Author name	Contains all authors from all references in the database records.
[ECNO]	EC/RN number	Number assigned by the Enzyme Commission or Chemical Abstract Service to designate a particular enzyme or chemical, respectively.
[FKEY]	Feature key	Contains the biological features assigned or annotated to the nucleotide sequences. Not available for the protein or structure databases.
[GENE]	Gene name	Contains the standard and common names of genes found in the database records.
[JOUR]	Journal name	Contains the name of the journal in which the data were published.
[KYWD]	Keyword	Contains special index terms from the controlled vocabularies associated with the GenBank, EMBL, DDBJ, SWISS-Prot, PIR, PRF, or PDB databases.
[MDAT]	Modification date	Contains the date that the most recent modification to that record is indexed in Entrez, in the format YYYY/MM/DD.
[MOLWT]	Molecular weight	Molecular weight of a protein, in daltons (Da), calculated by the method described in the Searching by Molecular Weight section of the Entrez help document.
[ORGN]	Organism	Contains the scientific and common names for the organisms associated with protein and nucleotide sequences.
[PROP]	Properties	Contains properties of the nucleotide or protein sequence. For example, the nucleotide database's properties index includes molecule types, publication status, molecule locations, and GenBank divisions.
[PROT]	Protein name	Contains the standard names of proteins found in database records.
[PDAT]	Publication date	Contains the date that records are released into Entrez, in the format YYYY/MM/DD.
[SQID]	SeqID	Contains the special string identifier for a given sequence.
[SLEN]	Sequence length	Contains the total length of the sequence.
[WORD]	Text word	Contains all of the "free text" associated with a record.
[TITL]	Title word	Includes only those words found in the definition line of a record.

Note: Some of these qualifiers are interchangeable with PubMed qualifiers.
Source: www.ncbi.nlm.nih.gov/entrez/query/static/help/helpdoc.html.

```
>gi|18203677|sp|Q9ZGE9|BCHN
MERVERENGCFHTFCPIASVAWLHRKIKDSFFLIVGTHTCAHFIQTALDVMVYAHSRFGFAVLEESDLVS
ASPTEELGKVVQQVVDEWHPKVIFVLSTCSVDILKMDLEVSCKDLSTRFGFPVLPASTSGIDRSFTQGED
AVLHALLPFVPKEAPAVEPVEEKKPRWFSFGKESEKEKAEPARNLVLIGAVTDSTIQQLQWELKQLGLPK
VDVFPDGDIRKMPVINEQTVVVPLQPYLNDTLATIRRERRAKVLSTVFPIGPDGTARFLEAICLEFGLDT
SRIKEKEAQAWRDLEPQLQILRGKKIMFLGDNLLELPLARFLTSCDVQVVEAGTPYIHSKDLQQELELLK
ERDVRIVESPDFTKQLQRMQEYKPDLVVAGLGICNPLEAMGFTTAWSIEFTFAQIHGFVNAIDLIKLFTK
PLLKRQALMEHGWAEAGWLE
```

Figure 2.4: Example of a FASTA file.

sequence analysis programs. The plain sequence in standard one-letter symbols starts in the second line. Each line of sequence data is limited to sixty to eighty characters in width. The drawback of this format is that much annotation information is lost.

Abstract Syntax Notation One. Abstract Syntax Notation One (ASN.1) is a data mark-up language with a structure specifically designed for accessing relational databases. It describes sequences with each item of information in a sequence record separated by tags so that each subportion of the sequence record can be easily added to relational tables and later extracted (Fig. 2.5). Though more difficult for people to read, this format makes it easy for computers to filter and parse the data. This format also facilitates the transimission and integration of data between databases.

Conversion of Sequence Formats

In sequence analysis and phylogenetic analysis, there is a frequent need to convert between sequence formats. One of the most popular computer programs for sequence format conversion is *Readseq*, written by Don Gilbert at Indiana University. It recognizes sequences in almost any format and writes a new file in an alternative format. The web interface version of the program can be found at: http://iubio.bio.indiana.edu/cgi-bin/readseq.cgi/.

SRS

Sequence retrieval system (SRS; available at http://srs6.ebi.ac.uk/) is a retrieval system maintained by the EBI, which is comparable to NCBI Entrez. It is not as integrated as Entrez, but allows the user to query multiple databases simultaneously, another good example of database integration. It also offers direct access to certain sequence analysis applications such as sequence similarity searching and Clustal sequence alignment (see Chapter 5). Queries can be launched using "Quick Text Search" with only one query box in which to enter information. There are also more elaborate submission forms, the "Standard Query Form" and the "Extended Query Form." The standard form allows four criteria (fields) to be used, which are linked by Boolean operators. The extended form allows many more diversified criteria and fields to be used. The search results contain the query sequence and sequence annotation as well as links to literature, metabolic pathways, and other biological databases.

```
          name "Tracking molecular evolution of photosynthesis by
characterization of a major photosynthesis gene cluster from Heliobacillus
mobilis." } ,
        authors {
          names
            std {
              {
                name
                  name {
                    last "Xiong" ,
                    initials "J." } } ,
              {
                name
                  name {
                    last "Inoue" ,
                    initials "K." } } ,
              {
                name
                  name {
                    last "Bauer" ,
                    initials "C.E." } } } ,
          affil
            str "Department of Biology, Indiana University, Bloomington, IN
47405, USA." } ,
        from
          journal {
            title {
              iso-jta "Proc. Natl. Acad. Sci. U.S.A." ,
              ml-jta "Proc Natl Acad Sci U S A" ,
              issn "0027-8424" ,
              name "Proceedings of the National Academy of Sciences of the
United States of America." } ,
            imp {
              date
                std {
                  year 1998 ,
                  month 12 ,
                  day 8 } ,
              volume "95" ,
              issue "25" ,
              pages "14851-14856" ,
              language "eng" } } ,
        ids {
          pubmed 9843979 ,
          medline 99061957 } } ,
      pmid 9843979 } ,
    comment "SEQUENCE FROM N.A." } } ,
 inst {
   repr raw ,
   mol aa ,
   length 440 ,
   seq-data
     ncbieaa "MERVERENGCFHTFCPIASVAWLHRKIKDSFFLIVGTHTCAHFIQTALDVMVYAHSRFGFAVL
EESDLVSASPTEELGKVVQQVVDEWHPKVIFVLSTCSVDILKMDLEVSCKDLSTRFGFPVLPASTSGIDRSFTQGEDA
VLHALLPFVPKEAPAVEPVEEKKPRWFSFGKESEKEKAEPARNLVLIGAVTDSTIQQLQWELKQLGLPKVDVFPDGDI
RKMPVINEQTVVVPLQPYLNDTLATIRRERRAKVLSTVFPIGPDGTARFLEAICLEFGLDTSRIKEKEAQAWRDLEPQ
LQILRGKKIMFLGDNLLELPLARFLTSCDVQVVEAGTPYIHSKDLQQELELLKERDVRIVESPDFTKQLQRMQEYKPD
LVVAGLGICNPLEAMGFTTAWSIEFTFAQIHGFVNAIDLIKLFTKPLLKRQALMEHGWAEAGWLE" } ,
```

Figure 2.5: A portion of a sequence file in ASN.1 format.

SUMMARY

Databases are fundamental to modern biological research, especially to genomic studies. The goal of a biological database is two fold: information retrieval and knowledge discovery. Electronic databases can be constructed either as flat files, relational, or object oriented. Flat files are simple text files and lack any form of organization to facilitate information retrieval by computers. Relational databases organize data as tables and search information among tables with shared features. Object-oriented databases organize data as objects and associate the objects according to hierarchical relationships. Biological databases encompass all three types. Based on their content, biological databases are divided into primary, secondary, and specialized databases. Primary databases simply archive sequence or structure information; secondary databases include further analysis on the sequences or structures. Specialized databases cater to a particular research interest. Biological databases need to be interconnected so that entries in one database can be cross-linked to related entries in another database. NCBI databases accessible through Entrez are among the most integrated databases. Effective information retrieval involves the use of Boolean operators. Entrez has additional user-friendly features to help conduct complex searches. One such option is to use Limits, Preview/Index, and History to narrow down the search space. Alternatively, one can use NCBI-specific field qualifiers to conduct searches. To retrieve sequence information from NCBI GenBank, an understanding of the format of GenBank sequence files is necessary. It is also important to bear in mind that sequence data in these databases are less than perfect. There are sequence and annotation errors. Biological databases are also plagued by redundancy problems. There are various solutions to correct annotation and reduce redundancy, for example, merging redundant sequences into a single entry or store highly redundant sequences into a separate database.

FURTHER READING

Apweiler, R. 2000. Protein sequence databases. *Adv. Protein Chem*. 54:31–71.

Blaschke, C., Hirschman, L., and Valencia, A. 2002. Information extraction in molecular biology. *Brief. Bioinform*. 3:154–65.

Geer, R. C., and Sayers, E. W. 2003. Entrez: Making use of its power. *Brief. Bioinform*. 4:179–84.

Hughes, A. E. 2001. Sequence databases and the Internet. *Methods Mol. Biol*. 167:215–23.

Patnaik, S. K., and Blumenfeld, O. O. 2001. Use of on-line tools and databases for routine sequence analyses. *Anal. Biochem*. 289:1–9.

Stein, L. D. 2003. Integrating biological databases. *Nat. Rev. Genet*. 4:337–45.