An Introduction to Software Tools for Biological Applications

Developing

Bioinformatics Computer Skills

O'REILLY®

Cynthia Gibas & Per Jambeck

Usage: `gzip -[options] filenames`

As usual, in addition to the standard Unix *compress*, there's a faster and more efficient GNU utility: *gzip*. *gzip* behaves in much the same way as *compress*, except that it gets better compression on average, since it uses a superior algorithm. *gzip* adds the suffix *.gz* to a file that it compresses. It emulates the *compress* options described earlier and adds a few of its own:

-N

(default setting) Preserves the original name and timestamp from the file being compressed

-q

(quiet mode) Suppresses warnings when running

-d

Returns a file that has been compressed by *gzip* to its uncompressed state; *gzip* can also recognize and uncompress files produced by *compress*

Part III: Tools for Bioinformatics

Chapter 6. Biological Research on the Web

The Internet has completely changed the way scientists search for and exchange information. Data that once had to be communicated on paper is now digitized and distributed from centralized databases. Journals are now published online. And nearly every research group has a web page offering everything from reprints to software downloads to data to automated data-processing services.

A simple web search for the word *bioinformatics* yields tens of thousands of results. The information you want may be number 345 in the list or it may not be found at all. Where can you go to find only the useful software and data, and scientific articles? You won't always get there by a simple web search. How can you judge which information is useful? Publication on the Web gives information an appearance of authority it may not merit. How can you judge if software will give the type of results you need and perform its function correctly?

In this chapter we examine the art of finding information on the Web. We cover search engines and searching, where to find scientific articles and software, and how to use the classic online information sources such as PubMed. And once you've located your information, we help you figure out how to use it. Among the largest sources of information for biologists are the public biological databases. We discuss the history of the public databases, data annotation, the various forms the data can take, and how to get data in and out. Finally, we give you some pointers on how to judge the quality of the information you find out there.

The Internet is a tremendously useful information source for biological research. In addition to allowing researchers to exchange software and data easily, it can be a source of the kind of practical advice about computer software and hardware, experimental methods and protocols, and laboratory equipment that you once could get only by buying a beer for a seasoned lab worker or computer hacker. Use the Internet, but use it wisely.

6.1 Using Search Engines

AltaVista, Lycos, Google, HotBot, Northern Light, Dogpile, and dozens of other search engines exist to help you find your way around the billion or more pages that make up the Web. As a scientist, however, you're not looking for common web commodities such as places to order books on the Web or online news or porn sites. You're looking for perhaps a couple of needles in a large haystack.

Knowing how to structure a query to weed out the majority of the junk that will come up in a search is very useful, both in web searching and in keyword-based database searching. Understanding how to formulate boolean queries that limit your search space is a critical research skill.

6.1.1 Boolean Searching

Most web surfers approach searching haphazardly at best. Enter a few keywords into the little box, and look at whatever results come up. But each search engine makes different default assumptions, so if you enter *protein structure* into Excite's query field, you are asking for an entirely different search than if you enter *protein structure* into Google's query field. In order to search effectively, you need to use boolean logic, which is an extremely simple way of stating how a group of things should be divided or combined into sets.

Search engines all use some form of boolean logic, as do the query forms for most of the public biological databases. Boolean queries restrict the results that are returned from a database by joining a series of search terms with the operators AND, OR, and NOT. The meaning of these operators is straightforward: joining two keywords with AND finds documents that contain only *keyword1* and *keyword2* ; using OR finds documents that contain either *keyword1* or *keyword2* (or both); and using NOT finds documents that contain *keyword1* but not *keyword2*.

However, search engines differ in how they interpret a space or an implied operator. Some search engines consider a space an OR, so when you type *protein structure*, you're really asking for protein or structure. If you search for protein *structure* on Excite, which defaults to OR, you come up with a lot of advertisements for fad diets and protein supplements before you ever get to the scientific sites you're interested in. On the other hand, Google defaults to AND, so you'll find only references that contain protein and structure, which is probably what you intended to look for in the first place. Find out how the search engine you're using works before you formulate your query.

Boolean queries are read from left to right, just like text. Parentheses can structure more complex boolean queries. For instance, if you look for documents that contain *keyword1* and one of either *keyword2* or *keyword3*, but not *keyword4*, your query would look like this: (*keyword1* AND (*keyword2* OR *keyword3*)) NOT *keyword4*.

Many search engines allow you to use quotation marks to specify a phrase. If you want to find only documents in which the words *protein structure* appear together in sequence, searching for "protein structure" is one way to narrow your results.

Let's say you want to search a literature database for references about computing electrostatic potentials for protein molecules, and you only want to look for references by two authors, Barry Honig and Andrew McCammon. You might structure a boolean query statement as follows:

```
((protein AND "electrostatic potential") AND (Honig OR McCammon))
```

This statement tells the search engine you want references that contain both the word protein and the phrase electrostatic potential, and that you require either one or the other of the names Honig and McCammon.

There are many excellent web tutorials available on boolean searching. Try a search with the phrase *boolean searching* in Google, and see what comes up.

6.1.2 Search Engine Algorithms

While the purpose of this book isn't to describe exhaustively how search engines work, there are significant differences in how search engines build their databases and rank sites. These differences make some search engines far more useful than others for searching science and technology web sites.

Key features to look at in a web search engine's database building and indexing strategies are free URL submission, full-text indexing, automated, comprehensive web crawling, a fast "refresh rate," and a sensible ranking strategy for results.

Our current favorite search engine is Google. Google is extremely comprehensive, indexing over 1 billion URLs. Pages are ranked based on how many times they are linked from other pages. Links from well-connected pages are considered more significant than links from isolated pages. The claim is that a Google search will bring you to the most well-traveled pages that match your search topic, and we've found that it works rather well. Google caches copies of web pages, so pages can be accessible even if the server is offline. It returns only pages that contain all the relevant query terms. Google uses a shorthand version of the standard boolean search formula, and it allows such specialized services as locating all the pages that link back to a page of interest.

For the neophyte user, however, HotBot is probably the best search engine. HotBot is relatively comprehensive and regularly updated, and it offers form-based query tools that eliminate the need for you to formulate even simple query statements.

6.2 Finding Scientific Articles

Scientists have traditionally been able to trust the quality of papers in print journals because these journals are refereed. An editor sends each paper to a group of experts who are qualified to judge the quality of the research described. These reviewers comment on the manuscript, often requiring additions, corrections, and even further experiments before the paper is accepted for publication. Print journals in the sciences are, increasingly frequently, publishing their content in an electronic format in addition to hardcopy. Almost every major journal has a web site, most of which are accessible only to subscribers, although access to abstracts usually is free. Scientific articles in these web journals go through the same process of review as their print counterparts.

Another trend is e-journals, which have no print counterpart. These journals are usually refereed, and it shouldn't be too hard to find out by whom. For instance, the *Journal of Molecular Modeling*, an electronic journal published by Springer-Verlag, has links to information about the journal's editorial policy prominently displayed on its home page.

An excellent resource for searching the scientific literature in the biological sciences is the free server sponsored by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine. This server makes it possible for anyone with a web browser to search the Medline database. There are other literature databases of comparable quality available, but most of these are not free. Your institution may offer access to such sources as Lexis-Nexis or Cambridge Scientific Abstracts.

Outside of refereed resources, however, anyone can publish information on the Web. Often research groups make papers available as technical reports on their web sites. These technical reports may never be peer reviewed or published outside the research group's home organization, and your only clue to their quality is the reputation and expertise of the authors. This isn't to say that you shouldn't trust or seek out these sources. Many government organizations and academic research groups have reference material of near-textbook quality on their web sites. For example, the University of Washington Genome Center has an excellent tutorial on genome sequencing, and NCBI has a good practical tutorial on use of the BLAST sequence alignment program and its variants.

6.2.1 Using PubMed Effectively

PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi) is one of the most valuable web resources available to biologists. Over 4,000 journals are indexed in PubMed, including most of the well-regarded journals in cell and molecular biology, biochemistry, genetics, and related fields, as well as many clinical publications of interest to medical professionals.

PubMed uses a keyword-based search strategy and allows the boolean operators AND, OR, and NOT in query statements. Users can specify which database fields to check for each search term by following the search term with a field name enclosed in square brackets.

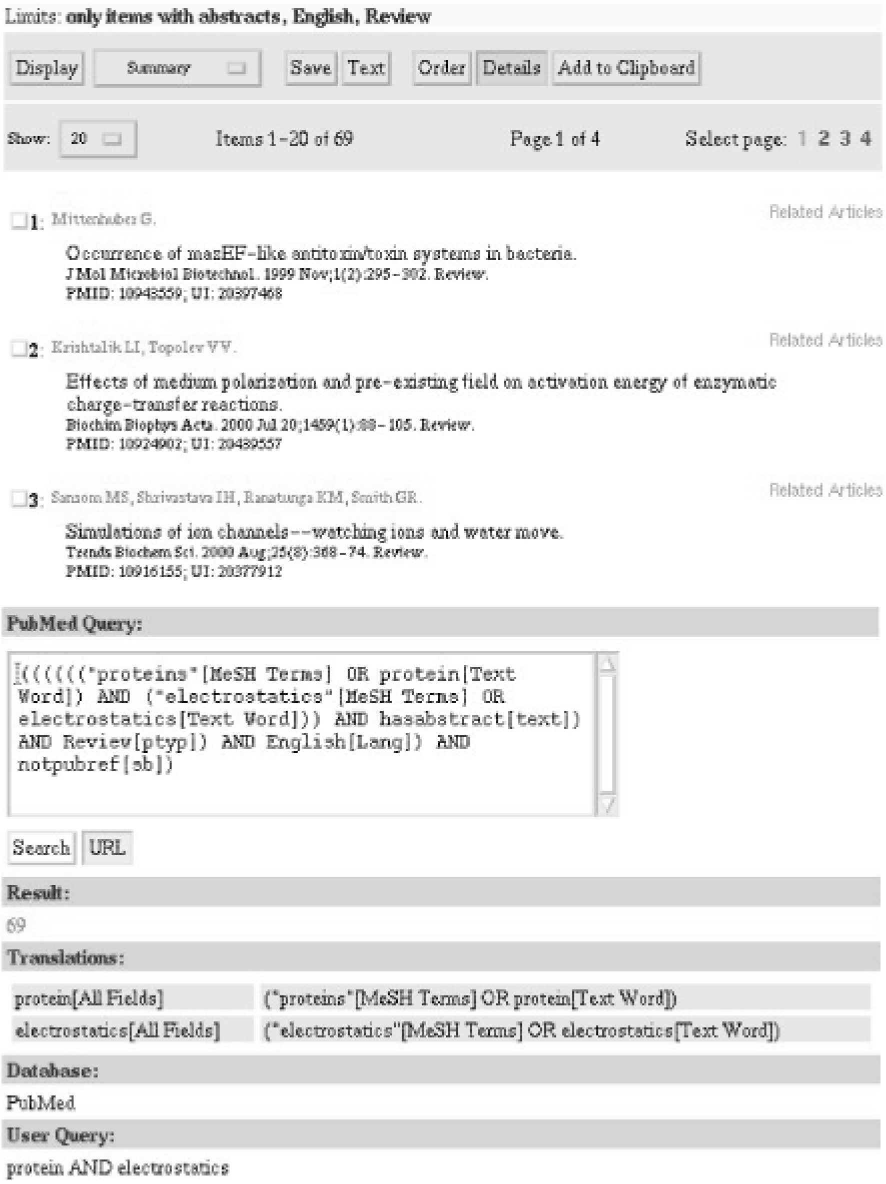
Additionally, users can search PubMed using Medical Subject Heading (MeSH) terms. MeSH is a library of standardized terms that may help locate manuscripts that use alternate terms to refer to the same concept. The MeSH browser (http://www.nlm.nih.gov/mesh/meshhome.html) allows users to enter a word or word fragment and find related keywords in the MeSH library. PubMed automatically finds MeSH terms related to query terms and uses them to enhance queries.

For example, we searched for "protein electrostatics" in PubMed. The terms protein and electrostatics are automatically joined with an AND unless otherwise specified. The resulting boolean query statement submitted to PubMed is actually:

```
((("proteins"[MeSH Terms] OR protein[Text Word]) AND
("electrostatics"[MeSH Terms]
 OR electrostatics[Text Word])) AND notpubref[sb])
```

The results of the search are shown in Figure 6-1.

Figure 6-1. Results from a PubMed search



Limits: **only items with abstracts, English, Review**

Items 1-20 of 69

Page 1 of 4

□1: Mittenhuber G.

Occurrence of mazEF-like antitoxin/toxin systems in bacteria.
J Mol Microbiol Biotechnol. 1999 Nov;1(2):295-302. Review.
PMID: 10943559; UI: 20397468

□2: Krishtalik LI, Topolev VV.

Effects of medium polarization and pre-existing field on activation energy of enzymatic charge-transfer reactions.
Biochim Biophys Acta. 2000 Jul 20;1459(1):88-105. Review.
PMID: 10924902; UI: 20439557

□3: Sansom MS, Shrivastava IH, Ranatunga KM, Smith GR.

Simulations of ion channels--watching ions and water move.
Trends Biochem Sci. 2000 Aug;25(8):368-74. Review.
PMID: 10916155; UI: 20377912

PubMed Query:

```
((((((("proteins"[MeSH Terms] OR protein[Text
Word]) AND ("electrostatics"[MeSH Terms] OR
electrostatics[Text Word])) AND hasabstract[text])
AND Review[ptyp]) AND English[Lang]) AND
notpubref[sb])
```

Result:

69

Translations:

protein[All Fields]	("proteins"[MeSH Terms] OR protein[Text Word])
electrostatics[All Fields]	("electrostatics"[MeSH Terms] OR electrostatics[Text Word])

Database:

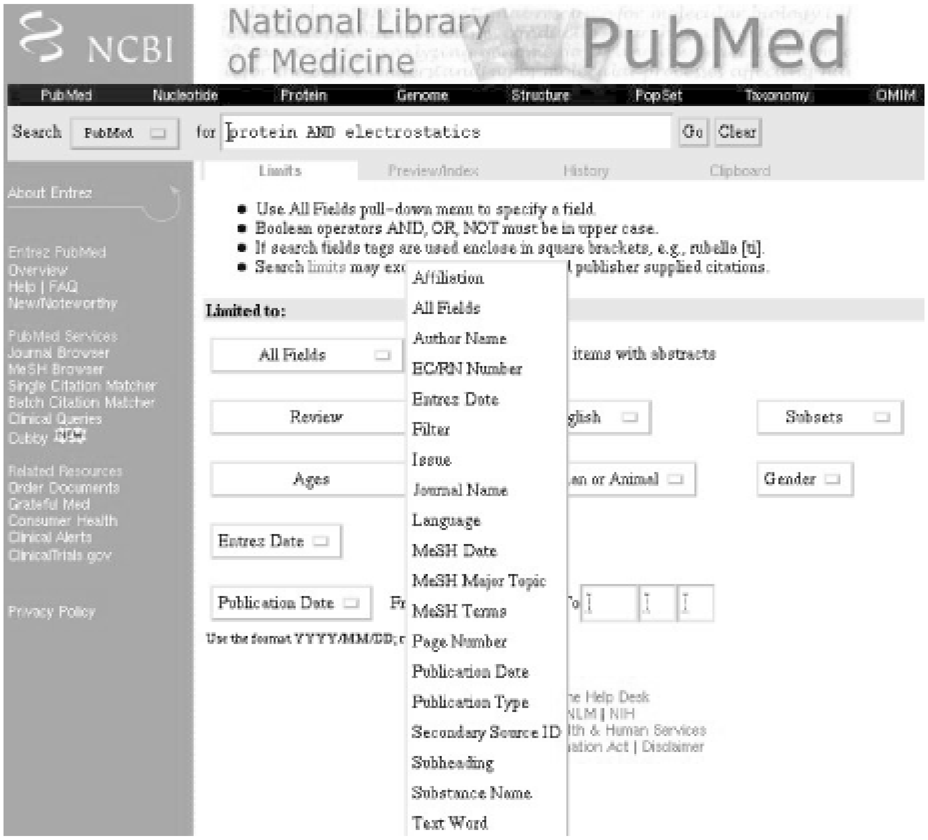
PubMed

User Query:

protein AND electrostatics

As you can see in Figure 6-2, PubMed also allows you to use a web interface to narrow your search. The Limits link immediately below the query box on the main PubMed page takes you to this web form.

Figure 6-2. Narrowing a search strategy using the Limits menu in PubMed



Search PubMed for protein AND electrostatics

- Use All Fields pull-down menu to specify a field.
- Boolean operators AND, OR, NOT must be in upper case.
- If search fields tags are used enclose in square brackets, e.g., rubella [ti].
- Search limits may exc… publisher supplied citations.

Limited to:

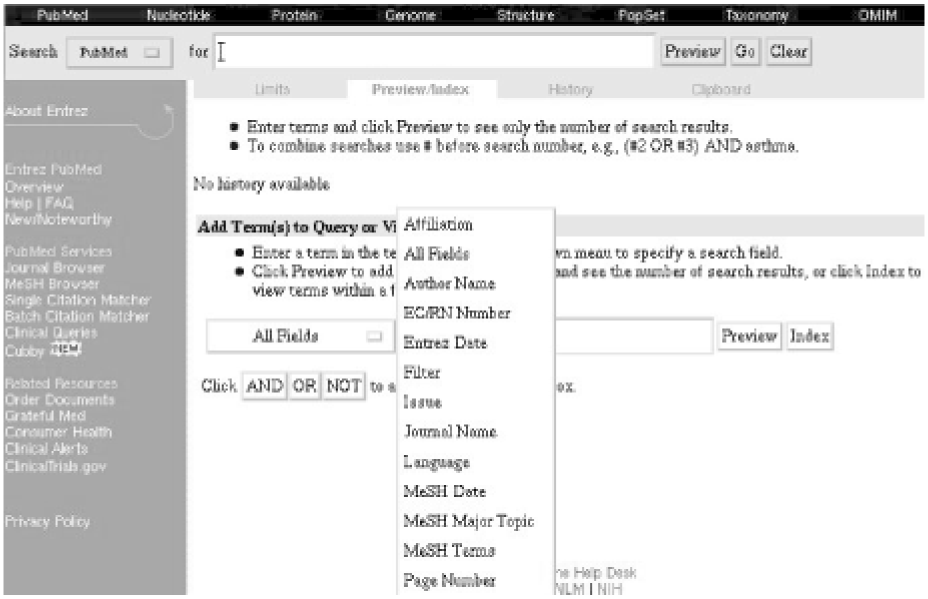
Use the format YYYY/MM/DD; …

The Limits form allows you to add specificity to your query. You can limit your search to particular fields in the PubMed database record, such as the Author Name or Substance Name field. Searches can also be limited by language, content (e.g., searching for review articles or clinical trials only), and date. For clinical research publications, the search can be limited based on the species, age, and gender of the research subjects.

The Preview/Index menu allows you to build a detailed query interactively. You can select a specific data field (for instance, the Author Name field) and then enter a term you want to search for within the specified field only. Clicking the AND, OR, or NOT buttons joins the new term to your previous query terms using the specified boolean operator.

For instance, you might start with a general search for "protein AND electrostatics," then go to the Preview/Index page (Figure 6-3) and specify that you want to search for "Gilson OR McCammon" in the Author Name field only.

Figure 6-3. Building a PubMed query using the Preview/Index form



- Enter terms and click Preview to see only the number of search results.
- To combine searches use # before search number, e.g., (#2 OR #3) AND asthma.

No history available

Add Term(s) to Query or Vi

- Enter a term in the te wn menu to specify a search field.
- Click Preview to add and see the number of search results, or click Index to view terms within a f

Click AND OR NOT to a ox.

You can also use the options in the History form to access results from earlier searches, and to narrow a search by adding new terms to the query.

If you want to collect results from multiple queries and save them into one big file, the Clipboard will allow you to do that. To save individual results to the Clipboard, simply click the checkbox next to the result you want to save, then click the Add to Clipboard button in the menu at the top of your results page.[1]

If you find a search strategy that works for you in PubMed, you can save that strategy in the form of a URL, and repeat the same search at any time in the future by visiting that URL. To save a PubMed URL, click the Details link on your results page, then click the URL link on the Details page. The URL of your search will appear in the Location field at the top of the web browser, so that you can bookmark it.

The "bookmarkable" URL for a PubMed search should look something like this:

```
http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?cmd=PureSearch&db=
PubMed&details_term=%28%28%28%28%28%28%22proteins%22%5BMeSH%20Terms
%5D%20OR%20protein%5BText%20Word%5D%29%20AND%20%28%22electrostatics
%22%5BMeSH%20Terms%5D%20OR%20electrostatics%5BText%20Word%5D%29%29
%20AND%20hasabstract%5Btext%5D%29%20AND%20Review%5Bptyp%5D%29%20AND
%20English%5BLang%5D%29%20AND%20notpubref%5Bsb%5D%29
```

Spending a few hours developing some detailed PubMed search strategies that work for you, and saving them, can save you a lot of work in the future.

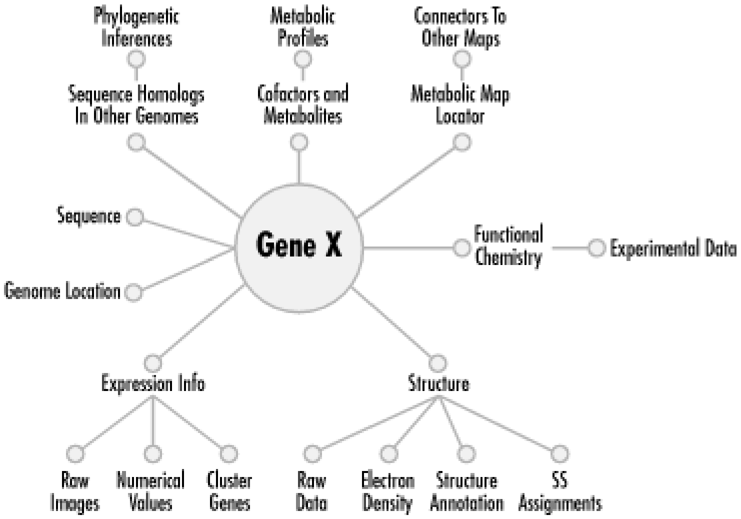
6.3 The Public Biological Databases

[1] You'll notice that all the checkbox-clicking to select and save individual results can get time-consuming if you're working with a lot of pages of results. It would be easier if you could come up with a search strategy that was absolutely certain to bring up only the results you want. There's no solution for this within the NCBI tools, and writing your own scripts to process batches of results may not help you either. The limitation is in the ability of computer programs to parse human language.

The nomenclature problem in biology at the molecular level is immense. Genes are commonly known by unsystematic names. These may come from developmental biology studies in model systems, so that some genes have names like *flightless*, *shaker*, and *antennapedia* due to the developmental effects they cause in a particular animal. Other names are chosen by cellular biologists and represent the function of genes at a cellular level, like *homeobox*. Still other names are chosen by biochemists and structural biologists and refer to a protein that was probably isolated and studied before the gene was ever found. Though proteins are direct products of genes, they are not always referred to by the same names or codes as the genes that encode them. This kind of confusing nomenclature generally means that only a scientist who works with a particular gene, gene product, or the biochemical process that it's a part of can immediately recognize what the common name of the gene refers to.

The biochemistry of a single organism is a more complex set of information than the taxonomy of living species was at the time of Linnaeus, so it isn't to be expected that a clear and comprehensive system of nomenclature will be arrived at easily. There are many things to be known about a given gene: its source organism, its chromosomal location, and the location of the activator sequences and identities of the regulatory proteins that turn it on and off. Genes also can be categorized by when during the organism's development they are turned on, and in which tissues expression occurs. They can be categorized by the function of their product, whether it's a structural protein, an enzyme, or a functional RNA. They can be categorized by the identity of the metabolic pathway that their product is part of, and by the substrate it modifies or the product it produces. They can be categorized by the structural architecture of their protein products. Clearly this is a wealth of information to be condensed into a reasonable nomenclature. Figure 6-4 shows a portion of the information that may be associated with a single gene.

Figure 6-4. Some of the information associated with a single gene



The problem for maintainers of biological databases becomes mainly one of annotation; that is, putting sufficient information into the database that there is no question of what the gene is, even if it does have a cryptic common name, and creating the proper links between that information and the gene sequence and serial number. Correct annotation of genomic data is an active research area in itself, as researchers attempt to find ways to transfer information across genomes without propagating error.

Storage of macromolecular data in electronic databases has given rise to a way of working around the problem of nomenclature. The solution has been to give each new entry into the database a serial number and then to store it in a relational database that knows the proper linkages between that serial number, any number of names for the gene or gene product it represents, and all manner of other information about the gene. This strategy is the one currently in use in the major biological databases. The questions databases resolve are essentially the same questions that arise in developing a nomenclature. However, by using relational databases and complex querying strategies, they (perhaps somewhat unfortunately) avoid the issue of finding a concise way for scientists to communicate the identities of genes on a nondigital level.

6.3.1 Data Annotation and Data Formats

The representation and distribution of biological data is still an open problem in bioinformatics. The nucleotide sequences of DNA and RNA and the amino acid sequences of proteins reduce neatly to character strings in which a single letter represents a single nucleotide or amino acid. The remaining challenges in representing sequence data are verification of the correctness of the data, thorough annotation of data, and handling of data that comes in ever-larger chunks, such as the sequences of chromosomes and whole genomes.

The standard reduced representation of the 3D structure of biomolecule consists of the Cartesian coordinates of the atoms in the molecule. This aspect of representing the molecule is straightforward. On the other hand, there are a host of complex issues for structure databases that are not completely resolved. Annotation is still an issue for structural data, although the biology community has attempted to form a consensus as to what annotation of a structure is currently required.

In the last 15 years, different researchers have developed their own styles and formats for reporting biological data. Biological sequence and structure databases have developed in parallel in the United States and in Europe. The use of proprietary software for data analysis has contributed a number of proprietary data formats to the mix. While there are many specialized databases, we focus here on the fields in which an effort is being made to maintain a comprehensive database of an entire class of data.

6.3.2 3D Molecular Structure Data

Though DNA sequence, protein sequence, and protein structure are in some sense just different ways of representing the same gene product, these datatypes currently are maintained as separate database projects and in unconnected data formats. This is mainly because sequence and structure determination methods have separate histories of development.

The first public molecular biology database, established nearly 10 years before the public DNA sequence databases, was the Protein Data Bank (PDB), the central repository for x-ray crystal structures of protein molecules.

While the first complete protein structure was published in the 1950s, there were not a significant number of protein structures available until the late 1970s. Computers had not developed to the point where graphical representation of protein structure coordinate data was possible, at least at useful speeds. However, in 1971, the PDB was established at the Brookhaven National Laboratory, to store protein structure data in a computer-based archive. A data format developed, which owed much of its style to the requirements of early computer technology. Throughout the 1970s and 1980s, the PDB grew. From 15 sets of coordinates in 1973, it grew to 69 entries in 1976. The number of coordinate sets deposited each year remained under 100 until 1988, at which time there were still fewer than 400 PDB entries.

Between 1988 and 1992, the PDB hit the turning point in its exponential growth curve. By January 1994, there were 2,143 entries in the PDB; at the time of this writing, the PDB has nearly reached the 14,000-entry mark. Management of the PDB has been transferred to a consortium of university and public-agency researchers, called the Research Collaboratory for Structural Bioinformatics, and a new format for recording of crystallographic data, the Macromolecular Crystallographic Information File (mmCIF), is being phased in to replace the antiquated PDB format. Journals that publish crystallographic results now require submission to the PDB as a condition of publication, which means that nearly all protein structure data obtained by academic researchers becomes available in the PDB in a fairly timely fashion.

A common issue for data-driven studies of protein structure is the redundancy and lack of comprehensiveness of the PDB. There are many proteins for which numerous crystal structures have been submitted to the database. Selecting subsets of the PDB data with which to work is therefore an important step in any statistical study of protein structure. As of December 1998, only about 2,800 of the protein chains in the PDB were sufficiently different from each other (having less than 95% of their sequence in common) to be considered unique. Many statistical studies of protein structure are based on sets of protein chains that have no more than 25% of their sequence in common; if this criterion is used, there are still only around 1,000 unique protein folds represented in the PDB. As the amount of biological sequence data available has grown, the PDB now lags far behind the gene-sequence databases.

6.3.3 DNA, RNA, and Protein Sequence Data

Sequence databases generally specialize in one type of sequence data: DNA, RNA, or protein. There are major sequence data collections and deposition sites in Europe, Japan, and the United States, and there are independent groups that mirror all the data collected in the major public databases, often offering some software that adds value to the data.

In 1970, Ray Wu sequenced the first segment of DNA; twelve bases that occurred as a single strand at the end of a circular DNA that was opened using an enzyme. However, DNA sequencing proved much more difficult than protein sequencing, because there is no chemical process that selectively cleaves the first nucleotide from a nucleic acid chain. When Robert Holley reported the sequencing of a 76-nucleotide RNA molecule from yeast, it was after seven years of labor. After Holley's sequence was published, other groups refined the protocols for sequencing, even successfully sequencing an 3,200-base bacteriophage genome. Real progress with DNA sequencing came after 1975, with the chemical cleavage method designed by

Allan Maxam and Walter Gilbert, and with Frederick Sanger's chain-terminator procedure.

The first DNA sequence database, established in 1979, was the Gene Sequence Database (GSDB) at Los Alamos National Lab. While GSDB has since been supplanted by the worldwide collaboration that is the modern GenBank, up-to-date gene sequence information is still available from GSDB through the National Center for Genome Resources.

The European Molecular Biology Laboratory, the DNA Database of Japan, and the National Institutes of Health cooperate to make all publicly available sequence data available through GenBank. NCBI has developed a standard relational database format for sequence data, known as the ASN.1 format. While this format promises to make locating the right sequences of the right kind in GenBank easier, there are still a number of services providing access to nonredundant versions of the database.

The DNA sequence database grew slowly through its first decade. In 1992, GenBank contained only 78,000 DNA sequences—a little over 100 million base pairs of DNA. In 1995, the Human Genome Project, and advances in sequencing technology, kicked GenBank's growth into high gear. GenBank currently doubles in size every 6 to 8 months, and its rate of increase is constantly growing.

6.3.4 Genomic Data

In addition to the Human Genome Project, there are now separate genome project databases for a large number of model organisms. The sequence content of the genome project databases is represented in GenBank, but the genome project sites also provide everything from genome maps to supplementary resources for researchers working on that organism. As of October 2000, NCBI's Entrez Genome database contained the partial or complete genomes of over 900 species. Many of these are viruses. The remainder include bacteria; archaea; yeast; commonly studied plant model systems such as *A. thaliana*, rice, and maize; animal model systems such as *C. elegans*, fruit flies, mice, rats, and puffer fish; as well as organelle genomes. NCBI's web-based software tools for accessing these databases are constantly evolving and becoming more sophisticated.

6.3.5 Biochemical Pathway Data

The most important biological activities don't happen by the action of single molecules, but as the orchestrated activities of multiple molecules. Since the early 20th century, biochemists have studied these functional ensembles of enzymes and their substrates. A few research groups have begun work on intelligently organizing and storing these pathways in databases. Two examples of pathway databases are WIT and KEGG. WIT, short for "What Is There?", was developed at Argonne National Labs. It's a database containing reconstructed metabolic pathways for organisms whose genomes have been entirely sequenced. The Kyoto Encyclopedia of Genes and Genomes (KEGG) stores similar data but links in information from sequence, structure, and genetic linkage databases. Both databases are queryable through web interfaces and are curated by a combination of automation and human expertise.

In addition to these whole genome "parts catalogs," other, more specialized databases that focus on specific pathways (such as intercellular signaling or degradation of chemical compounds by microbes) have been developed.

6.3.6 Gene Expression Data

DNA microarrays (or *gene chips*) are miniaturized laboratories for the study of gene expression. Each chip contains a deliberately designed array of probe molecules that can bind specific pieces of DNA or mRNA. Labeling the DNA or RNA with fluorescent molecules allows the level of expression of any gene in a cellular preparation to be measured quantitatively. Microarrays also have other applications in molecular biology, but their use in studying gene expression has opened up a new way of measuring genome functions.

Since the development of DNA microarray technology in the late 1990s, it has become apparent that the increase in available gene expression data will eventually parallel the growth of the sequence and structure databases, and that this is another datatype for which public access to raw data will be desirable. Raw microarray data has just begun to be made available to the public in selective databases, and talk of establishing a central data repository for such data is underway. However, formats for delivering this kind of data are still not standardized; often, it's made available in large spreadsheets or tab-delimited text. Two of the most comprehensive resources for microarray data are the National Human Genome Research Initiative's Microarray Project site and the Stanford Genome Resources site. Since many of the early microarray expression experiments were performed at Stanford, their genome resources site has links to both raw data and, in some cases, databases that can be queried using gene names or functional descriptions. Recently, the European Bioinformatics Institute has been instrumental in developing a set of standards for deposition of microarray data in databases. Several databases also exist for the deposition of 2D gel electrophoresis results, including SWISS-2DPAGE and HSC-2DPAGE. 2D-PAGE is a technology that allows quantitative study of protein concentrations in the cell, for many proteins simultaneously. The combination of these two techniques is a powerful tool for understanding how genomes work.

Table 6-1 summarizes sources on the Web for some of the most important databases we've discussed in this section.

Table 6-1. Major Biological Data and Information Sources

Subject	Source	Link
Biomedical literature	PubMed	http://www.ncbi.nlm.nih.gov/entrez/query.fcgi
Nucleic acid sequence	GenBank	http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?db=Nucleotide
	SRS at EMBL /EBI	http://srs.ebi.ac.uk
Genome sequence	Entrez Genome	http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?db=Genome
	TIGR databases	http://www.tigr.org/tdb/
Protein	GenBank	http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?db=Protein

sequence		
	SWISS-PROT at ExPASy	http://www.expasy.ch/spro/
	PIR	http://www-nbrf.georgetown.edu
Protein structure	Protein Data Bank	http://www.rcsb.org/pdb/
Entrez Structure DB		
Protein and peptide mass spectroscopy	PROWL	http://prowl.rockefeller.edu
Post-translational modifications	RESID	http://www-nbrf.georgetown.edu/pirwww/search/textresid.html
Biochemical and biophysical information	ENZYME	http://www.expasy.ch/enzyme/
	BIND	http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?db=Structure
Biochemical pathways	PathDB	http://www.ncgr.org/software/pathdb/
	KEGG	http://www.genome.ad.jp/kegg/
	WIT	http://wit.mcs.anl.gov/WIT2/
Microarray	Gene Expression Links	http://industry.ebi.ac.uk/~alan/MicroArray/
2D-PAGE	SWISS-2DPAGE	http://www.expasy.ch/ch2d/ch2d-top.html
Web resources	The EBI Biocatalog	http://www.ebi.ac.uk/biocat/
	IUBio Archive	http://iubio.bio.indiana.edu

6.4 Searching Biological Databases

There are dozens of biological databases on the Web, and many alternate web interfaces that provide access to the same sets of data. Which ones you use depends on your needs, but it's necessary for you to be aware of what the central data repositories are for various datatypes, and how often the more peripheral databases you might be using synchronize themselves with these central data sources.

Although data repositories for new types of biological data are multiplying, we focus here on two established databases: NCBI's GenBank, for DNA sequence data; and the Protein Data Bank, for molecular structure data. Every database has its own deposition procedures, and for the newer datatypes these are not yet well established or are still changing rapidly. However, both NCBI and RCSB have mature, automated, web-based deposition systems that are not likely to change drastically in the near future.

6.4.1 GenBank

NCBI, in cooperation with EMBL and other international organizations, provides the most complete collection of DNA sequence data in the world, as well as PubMed, a taxonomy database, and an alternate access point for protein sequence and structure data. This database, known as GenBank, may be accessed at http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?db=Protein.

NCBI maintains sequence data from every organism, every source, every type of DNA—from mRNA to cDNA clones to expressed sequence tags (ESTs) to high-throughput genome sequencing data and information about sequence polymorphisms. Users of the NCBI database need to be aware of the differences between these datatypes so that they can search the data set that's most appropriate for the work they're doing. The main sequence types that you'll encounter in a full GenBank search include:

mRNA

> Messenger RNA, the product of transcription of genomic DNA. mRNA may be edited by the cell to remove introns (in eukaryotes) or in other ways that result in differences from the transcribed genomic DNA. May be "partial" or "complete"; an mRNA may not cover the complete coding sequence of a gene.

cDNA

> A DNA sequence artificially generated by reverse transcription of mRNA. cDNA roughly represents the coding components of the genomic DNA region that produced the mRNA. May also be "partial" or "complete."

Genomic DNA

> A DNA sequence from genome sequencing that contains both coding and noncoding DNA sequences. May contain introns, repeat regions, and other features. Genomic DNA (as opposed to genome survey sequence) is generally "complete"; it's a result of multiple sequencing passes over a single stretch of a genome, and can generally be relied upon as a fairly good representation of the real DNA sequence of that region.

EST

> Short cDNA sequences prepared from mRNA extracted from a cell under particular conditions or in specific developmental phases (e.g., arabidopsis thaliana 2-week old shoots or valencia orange seeds). ESTs are used for quick identification of genes and don't cover the entire coding sequence of a gene.

GSS

Genome survey sequence. Single-pass sequence direct from the genome projects. Covers each region of sequence only once and is likely to contain a relatively large proportion of sequencing errors. You'd include genome survey sequence in a search only if you were looking for very new hypothetical gene annotations in a genome project that's still in progress.

There are two ways to search GenBank. The first is to use a text-based query to search the annotations associated with each DNA sequence entry in the database. The second, which we'll discuss in Chapter 7, is to use a method called BLAST to compare a query DNA (or protein) sequence to a sequence database.

Here's a sample GenBank record. Each GenBank entry contains annotation—information about the gene's identity, the conditions under which it was characterized, etc.—in addition to sequence.

```
LOCUS        AB009351 1412 bp   mRNA   PLN       22-JUN-1999
DEFINITION   Citrus sinensis mRNA for chalcone synthase, complete cds,
clone
             CitCHS2.
ACCESSION    AB009351 VERSION AB009351.1 GI:5106368
KEYWORDS     chalcone synthase.
SOURCE       Citrus sinensis young seed cDNA to mRNA, clone:CitCHS2.
  ORGANISM   Citrus sinensis
             Eukaryota; Viridiplantae; Streptophyta; Embryophyta;
Tracheophyta;
             euphyllophytes; Spermatophyta; Magnoliophyta;
eudicotyledons; core
             eudicots; Rosidae; eurosids II; Sapindales; Rutaceae;
Citrus.
REFERENCE    1 (sites)
  AUTHORS    Moriguchi,T., Kita,M., Tomono,Y., EndoInagaki,T. and
Omura,M.
  TITLE      One type of chalcone synthase gene expressed during
embryogenesis
             regulates the flavonoid accumulation in citrus cell
cultures
  JOURNAL    Plant Cell Physiol. 40 (6), 651-655 (1999)
  MEDLINE    99412624
  [...]
FEATURES     Location/Qualifiers
  Source     1..1412
             /organism="Citrus sinensis"
             /db_xref="taxon:2711"
             /clone="CitCHS2"
             /dev_stage="young seed"
             /note="Valencia orange"
  CDS        30..1205
             /codon_start=1
             /product="chalcone synthase"
             /protein_id="BAA81664.1"
             /db_xref="GI:5106369"
             /translation="MATVQEIRNAQRADGPATVLAIGTATPAHSVNQADYPDYYFRIT
             KSEHMTELKEKFKRMCDKSMIKKRYMYLTEEILKENPNMCAYMAPSLDARQDIVVVEV
             PKLGKEAATKAIKEWGQPKSKITHLIFCTTSGVDMPGADYQLTKLIGLRPSVKRFMMY
             QQGCFAGGTVLRLAKDLAENNKGARVLVVCSEITAVTFRGPADTHLDSLVGQALFGDG
             AAAVIVGADPDTSVERPLYQLVSTSQTILPDSDGAIDGHLREVGLTFHLLKDVPGLIS
             KNIEKSLSEAFAPLGISDWNSIFWIAHPGGPAILDQVESKLGLKGEKLKATRQVLSEY
             GNMSSACVLFILDEMRKKSVEEAKATTGEGLDWGVLFGFGPGLTVETVVLHSVPIKA"
  polyA_site 1412
             /note="18 a nucleotides"
BASE COUNT   331 a    358 c    372 g    351 t
ORIGIN
  1 aaacatattc attaagggtt caacttgaaa tggcaaccgt tcaagagatc agaaacgctc
  61 agcgtgccga cggcccggcc accgtcctcg ccatcggtac ggccacgcct gcccacagtg
  121 tcaaccaggc tgattatccc gactattact tcaggatcac aaagagcgag catatgacgg
  [...]
  1261 cacagttgag ttattggttg atcgtgtgaa ggtttagttt tgtcaattga
gtttaaggca
  1321 tcgtgccttt tctcttatga cgtcaccaaa cctgggcaac gctttgtgtt
tatgcataaa
  1381 ttcttgggaa tttgagaaag tagtaaattt gt
//
```

This sample GenBank record shows the types of fields that can be found in a record from the GenBank Nucleotide database. Everything from the identity of the protein product (in this example, chalcone synthase), the sequence of the protein product, and its starting and ending point within the gene, to the authors who submitted the record and the journal references in which the experiment was described, can be found in the record, and therefore can be used to search the database.

The GenBank search interface is nearly identical to the PubMed search interface. The Limits, Preview/Index, History, and Clipboard features for searching work the same way in the Protein, Nucleic Acid, and Genome databases as they do for PubMed, although the specific fields that can be searched and limits that can be set are somewhat different.

6.4.1.1 Saving search results

Sequences can be downloaded from NCBI in any of three file formats: the simple FASTA format, which is readable by many sequence analysis programs but contains little information other than sequence; the GenBank flat file format, which is a legacy flat file format that was used at GenBank earlier in its history; and the modern ASN.1 (Abstract Syntax Notation One) format. ASN.1 is a generic data specification, designed to promote database interoperability, that is now used for storage and retrieval of all datatypes—sequences, genomes, structure, and literature—at NCBI. The NCBI Toolkit, a code library for developing molecular biology software, relies on the ASN.1 specification. NCBI, and increasingly, other organizations, rely on the NCBI Toolkit for software development. Learning to use the NCBI Toolkit is a programming challenge well beyond the scope of this book, but there is an excellent tutorial on the Web, developed by Christopher Hogue and his research group at the Samuel Lunenfeld Research Institute.

The casual database user or depositor doesn't have to think too much about file formats, except if database files are to be exported and read by another piece of software. NCBI's forms-based interfaces convert user-entered data into the appropriate format for deposition, and the availability of GenBank files in FASTA format means that most sequence analysis software can handle sequence files you download from NCBI without complicated conversions.

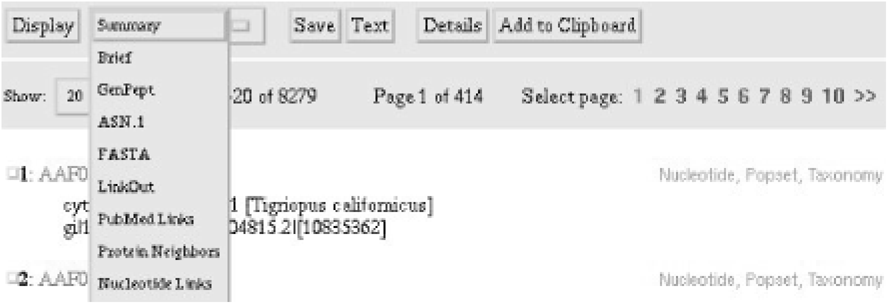
When you save results of a GenBank search, you can choose the format in which to save them. Earlier, you saw what the GenBank sequence record looks like. Many of the computer programs we discuss in the following chapters can read GenBank format sequence files, but some can't. A particularly foolproof format in which to save your sequence files if you're going to process them with other software is the FASTA format. FASTA files have a simple format, a single comment line that begins with a > character, followed by single-character DNA sequence on as many lines as needed to hold the sequence, with no breaks. Of course, some information associated with the gene is lost when you save the data in FASTA format, but if the program you want to use can't read that extra data, it won't be useful to you anyway.

Here's a sample of data in FASTA format:

```
> gene identifier and comments here
MATVQEIRNAQRADGPATVLAIGTATPAHSVNQADYPDYYFRITKSEHMTELKEKFKRMCDKSMIKKRYM
YLTEEILKENPNMCAYMAPSLDARQDIVVVEVPKLGKEAATKAIKEWGQPKSKITHLIFCTTSGVDMPGA
DYQLTKLIGLRPSVKRFMMYQQGCFAGGTVLRLAKDLAENNKGARVLVVCSEITAVTFRGPADTHLDSLV
GQALFGDGAAAVIVGADPDTSVERPLYQLVSTSQTILPDSDGAIDGHLREVGLTFHLLKDVPGLISKNIE
KSLSEAFAPLGISDWNSIFWIAHPGGPAILDQVESKLGLKGEKLKATRQVLSEYGNMSSACVLFILDEMR
KKSVEEAKATTGEGLDWGVLFGFGPGLTVETVVLHSVPIKA
```

To save your files in FASTA format, simply use the pulldown menu at the top of the results page. When you first see it, it will say "Summary," but you can change it to FASTA, ASN.1, and other formats. Once you've chosen your format, you can click the Save button to save all your sequences into one big FASTA-format file. Figure 6-5 shows you how to change the file formats when doing a GenBank search.

Figure 6-5. Changing the file format to write out your GenBank search results



6.4.1.2 Saving large result sets

So far, our discussion of information retrieval from databases has assumed that you need access to only a few sequences at a time. However, modern bioinformatics studies increasingly deal with large amounts of sequence data. For example, genefinding programs (covered in Chapter 7) are trained and tested on hundreds or thousands of DNA sequences; comprehensive studies of protein families can involve analysis of up to thousands of protein sequences as well. While it's possible to select thousands of checkboxes on a web page by hand, it would be better to use an automated tool that can return a large number of sequences based on criteria you specify.

NCBI provides just such a tool in the form of Batch Entrez (http://www.ncbi.nlm.nih.gov/Entrez/batch.html). Batch Entrez is one of the tools accessible from the Entrez web site. It's accessed using a web form that allows the user to select sequences by source organism, by an Entrez query (using the query structure described in the section on PubMed), or by a list of accession numbers (provided by the user in the form of a text file). The results of a Batch Entrez search are then packaged in a file that is downloaded to the user's computer, where the complete result set can be edited manually or (even better) using a script.

At this time, not all the biological databases are so kind about providing such services, but all the public databases have FTP sites that allow you to download the entire database in one form or another. That can take up a lot of space on your hard disk, but disk space is cheaper these days than the time it would take you to handle a large set of results on an interactive web site. If you've got a local copy of the big databases that interest you, you can write (or perhaps even download) a script that processes the database, looking for your keyword of choice, and writes out the information you want to a file.

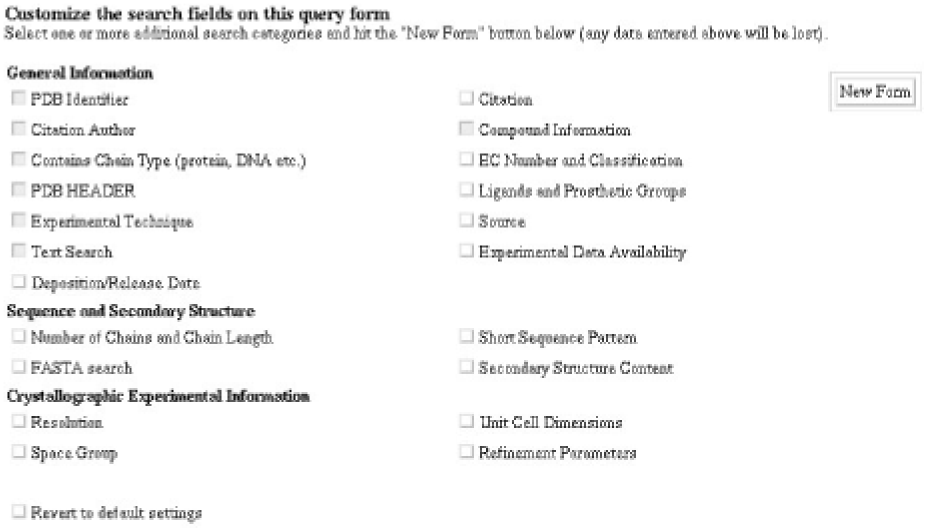
6.4.2 PDB

Unlike NCBI, the Protein Data Bank (http://www.rcsb.org/pdb/) is responsible for only one type of molecular data: molecular structures of molecules and, to a growing extent, the underlying raw data sets from which the molecular structures were modeled.

The PDB web site offers three options for searching the database. You can enter a four-letter PDB identifier directly, or search using the SearchLite or SearchFields interfaces. The SearchLite interface is similar to the other query tools we've discussed. You can enter a term or terms into the query box, joined by the operators AND, OR, and BUTNOT.

The SearchFields interface is an innovative design-it-yourself web form system. As you see in Figure 6-6, when you first go to SearchFields, you can scroll down to the bottom of the web form and select which parts of the form you need. If you're only going to be doing a FASTA search to find similar sequences, you don't need a search form that prompts you for keywords to use in searching the Citation Author field. You might want to add a field that lets you search for proteins with a particular ligand or prosthetic group. With the SearchFields interface, you select the form elements you want for your custom PDB search, and click the "New Form" button to generate the new query form.

Figure 6-6. Customizing the PDB's SearchFields form



Customize the search fields on this query form
Select one or more additional search categories and hit the "New Form" button below (any data entered above will be lost).

General Information

- □ PDB Identifier
- □ Citation Author
- □ Contains Chain Type (protein, DNA etc.)
- □ PDB HEADER
- □ Experimental Technique
- □ Text Search
- □ Deposition/Release Date
- □ Citation
- □ Compound Information
- □ EC Number and Classification
- □ Ligands and Prosthetic Groups
- □ Source
- □ Experimental Data Availability

Sequence and Secondary Structure

- □ Number of Chains and Chain Length
- □ FASTA search
- □ Short Sequence Pattern
- □ Secondary Structure Content

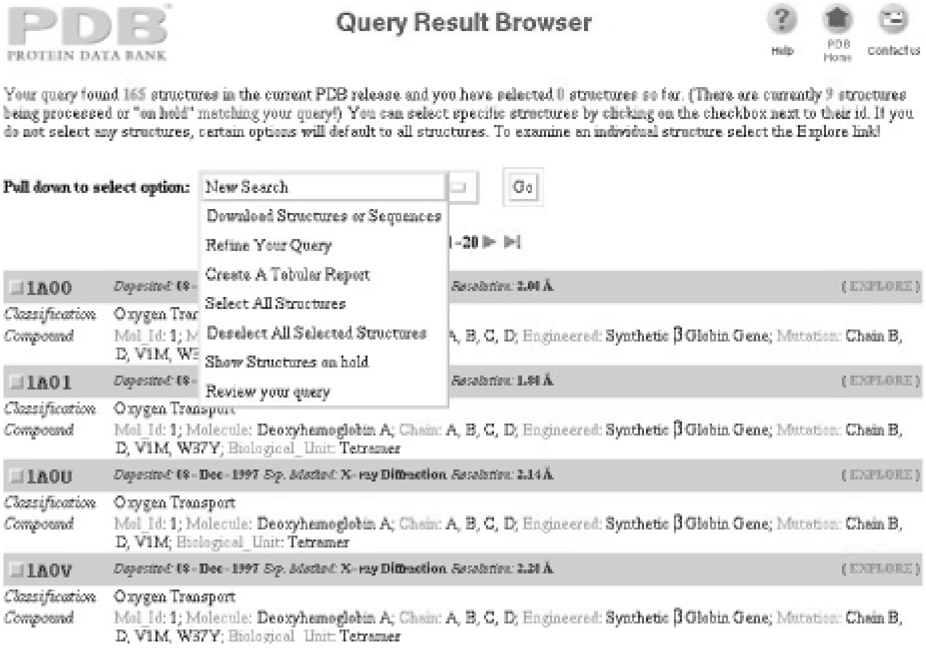
Crystallographic Experimental Information

- □ Resolution
- □ Space Group
- □ Unit Cell Dimensions
- □ Refinement Parameters

□ Revert to default settings

Whether you use SearchLite or SearchFields, you'll come to the Query Result browser (Figure 6-7), where you can select options for refining your query, downloading your results as structure or sequence files, and even preparing a tabular report of your search results. These options are straightforward to use and well documented on the PDB web site.

Figure 6-7. Options for using query results at the PDB



Your query found 165 structures in the current PDB release and you have selected 0 structures so far. (There are currently 9 structures being processed or "on hold" matching your query!) You can select specific structures by clicking on the checkbox next to their id. If you do not select any structures, certain options will default to all structures. To examine an individual structure select the Explore link!

Pull down to select option: New Search

□ **1A00** *Deposited:* 08- ... *Resolution:* 2.00 Å
Classification Oxygen Tra...
Compound Mol_Id: 1; M... A, B, C, D; Engineered: Synthetic β Globin Gene; Mutation: Chain B, D, V1M, W...

□ **1A01** *Deposited:* 08- ... *Resolution:* 1.80 Å
Classification Oxygen Transport
Compound Mol_Id: 1; Molecule: Deoxyhemoglobin A; Chain: A, B, C, D; Engineered: Synthetic β Globin Gene; Mutation: Chain B, D, V1M, W37Y; Biological_Unit: Tetramer

□ **1A0U** *Deposited:* 08-Dec-1997 *Exp. Method:* X-ray Diffraction *Resolution:* 2.14 Å
Classification Oxygen Transport
Compound Mol_Id: 1; Molecule: Deoxyhemoglobin A; Chain: A, B, C, D; Engineered: Synthetic β Globin Gene; Mutation: Chain B, D, V1M; Biological_Unit: Tetramer

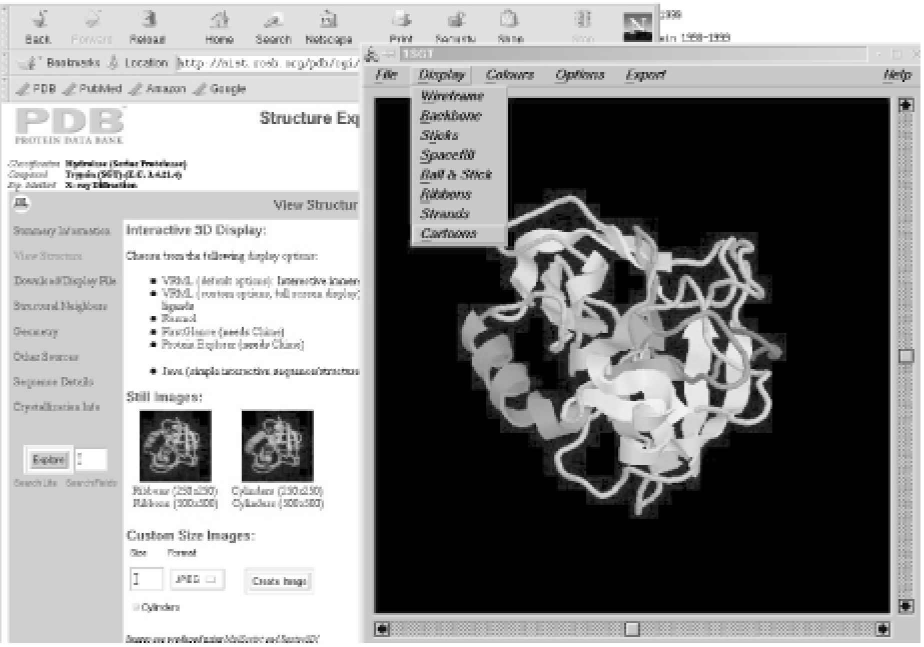
□ **1A0V** *Deposited:* 08-Dec-1997 *Exp. Method:* X-ray Diffraction *Resolution:* 2.20 Å
Classification Oxygen Transport
Compound Mol_Id: 1; Molecule: Deoxyhemoglobin A; Chain: A, B, C, D; Engineered: Synthetic β Globin Gene; Mutation: Chain B, D, V1M, W37Y; Biological_Unit: Tetramer

The Protein Data Bank makes data available in two formats: the legacy PDB flat- file format, and the newer mmCIF data format. We'll discuss the differences between these two file formats in more detail in Chapter 12. At this point, little of the available structure-analysis and protein-modeling software handles the mmCIF format, so you are not likely to need to download protein structure data in mmCIF format unless you are developing new software.[2] You can choose to download the complete set of results from your search as a *tar* archive or a zipped file in either PDB or mmCIF format, as well as in sequence-only FASTA format.

[2] The PDB offers a suite of mmCIF and PDB format conversion tools, as well as code libraries for working with mmCIF files.

Another convenient way to view protein structure data from the PDB web site is to install a browser plug-in such as RasMol or Chime on your computer. We discuss how to do this in Chapter 9. Once the plug-in is installed and properly configured, you can simply click on a link on the protein's View Structure page and the protein structure is automatically displayed using the plug-in, as shown in Figure 6-8.

Figure 6-8. Viewing a PDB file using a browser plug-in



Location: http://nist.rcsb.org/pdb/cgi/

Classification: Hydrolase (Serine Proteinase)
Compound: Trypsin (SGT) (E.C. 3.4.21.4)
Exp. Method: X-ray Diffraction

Interactive 3D Display:

Choose from the following display options:

- VRML (default options): Interactive lowres
- VRML (custom options, full screen display, ligands
- Rasmol
- FirstGlance (needs Chime)
- Protein Explorer (needs Chime)
- Java (simple interactive sequence/structure

Still Images:

Ribbons (250x250) Ribbons (500x500)
Cylinders (250x250) Cylinders (500x500)

Custom Size Images:

Size Format JPEG

6.5 Depositing Data into the Public Databases

In addition to downloading information from the public databases, you may also submit your own results.

6.5.1 GenBank Deposition

Deposition of sequences to GenBank has been made extremely simple by NCBI. Users depositing only a few sequences can use the web-based BankIt tool, which is a self-explanatory form-based interface accessible from the GenBank main page at NCBI. Users submitting multiple sequences or other complicated submissions can use NCBI's Sequin software, which is available for all major operating systems. Sequin is well documented on the NCBI site. NCBI has recently established two special submission paths: EST sequences should be submitted through dbEST, rather than to GenBank, and genome survey sequences through dbGSS.

6.5.2 PDB Deposition

Deposition of structures to the PDB are done using the AutoDep input tool (ADIT). AutoDep is a tool that integrates data validation software with the deposition process so that the user can receive feedback on data quality during the deposition process. AutoDep is tied in with the curation tools the PDB uses to prepare structure data for inclusion in the data bank.

6.6 Finding Software

Bioinformatics is a diffuse field, attracting researchers from many disciplines, and articles about new research developments in bioinformatics are widely distributed in the literature. If you're looking for cutting-edge developments, journals such as *Bioinformatics*, *Nucleic Acids Research*, *Journal of Molecular Biology,* and *Protein Science* often publish papers describing innovations in computational biology methods.

If you're looking for proven software for a particular application, there are a number of reliable web resource lists that link to computational biology software sites. Most of the major biological databases have software resource listings and the necessary motivation to keep their listings up-to-date. The PDB links to the best free software packages for macromolecular structure refinement, visualization, and dynamics. TIGR and NCBI provide links to many tools for protein and DNA sequence analysis.

Many organizations and groups provide web implementations of their software. These can be a great time-saver, especially if you are new to the use of noncommercial software packages in research. Many of the bioinformatics programs that we describe in this book are also available as web servers. You can use the web-server versions to get you started and understand the inputs, outputs, and options for the program. However, web servers have their drawbacks. They typically implement only the most popular options in any software package: it's difficult to design a web form that allows you to select every option in a complicated program. They often allow you to run only one calculation at a time. This is fine if you're only interested in analyzing a few sequences or structures, and not so fine if you suddenly find yourself with 500 sequences to analyze.

With a little clever programming, you can develop scripts that allow you to hit a web server with multiple requests without entering them manually into a form, but if you're capable of doing that, you're probably able to download a local copy of the software and run it on your own machine. Using your own processor in such cases avoids slow data transfer to and from remote sites and is also considered more polite than running huge jobs on someone else's web server.

In the next four chapters, we'll discuss the software packages you are most likely to want to use. We'll show you how to set them up on your own computer and use them independent of web interfaces.

We can't cover every available software package and web server in this book; there are just too many. You will eventually want to go out on your own and find new tools to use. Keep a few things in mind when searching for software, and you'll soon be able to judge for yourself if a new computer program is something you want to use.

Ideally, you have access to the source code (the human-readable version of a computer program) for whatever the web server is doing, and you can read the source code and know it's doing what you expect. But you might not know how to read source code, and even if you do, you might not be able to get hold of it. Unfortunately, some bioinformatics software authors don't make their source code publicly available, preferring to set up web servers that are easier to use and maintain. This can incidentally have the effect of hiding the underlying method from close scrutiny by users.

If you can't read the source code, what can you read? Most software or web servers made available by academic researchers or government institutions have online help pages and other documentation, including bibliographic information for publications in refereed journals that describe the methods encoded in the software. Read this documentation and understand the method and its results before you use it, just as you would for an experimental method that is new to you.

If the program or server you want to use has no documentation and doesn't allow you to check the source code, you should seriously consider not using that program, unless you have some way to verify its output (for instance, by comparison with the output of a well-documented program). After all, you're drawing conclusions based on your results; do you want to stake your scientific credibility on an unknown quantity?

6.7.3 Timeliness

One of the most frequently linked biology resource sites on the Web is Pedro's Biomolecular Research Tools (http://www.public.iastate.edu/~pedro/research_tools.html). Sites all over the world still have pointers to this collection of links. And yet, if you click to Pedro's site, you'll find that the collection was last updated in 1996. A funny thing about the Web is that out-of-date sites don't just go away. They remain on the server, looking authoritative. Check web sites for dates. If there's no sign of activity in or reference to the current year, be skeptical.

Timeliness isn't always an issue with software. Software written in 1980 can be as useful and functional now as it was then. What you may encounter are problems compiling software that incorporates proprietary technologies that are no longer supported, or code libraries that have since ceased to be developed.

Chapter 7. Sequence Analysis, Pairwise Alignment, and Database Searching

We now begin our tour of bioinformatics tools in earnest. In the next five chapters, we describe some of the software tools and applications you can expect to see in current research in computational biology. From gene sequences to the proteins they encode to the complicated biological networks they are involved in, computational methods are available to help you analyze data and formulate hypotheses. We have focused on commonly used software packages and packages we have used; to attempt to encompass every detail of every program out there, however, we'd need to turn every chapter in this book into a book of its own.

The first tools we describe are those that analyze protein and DNA sequence data. Sequence data is the most abundant type of biological data available electronically. While other databases may eventually rival them in size, the importance of sequence databases to biology remains central. Pairwise sequence comparison, which we discuss in this chapter, is the most essential technique in computational biology. It allows you to do everything from sequence-based database searching, to building evolutionary trees and identifying characteristic features of protein families, to creating homology models. But it's also the key to larger projects, limited only by your imagination—comparing genomes, exploring the sequence determinants of protein structure, connecting expression data to genomic information, and much more.

The types of analysis that you can do with sequence data are:

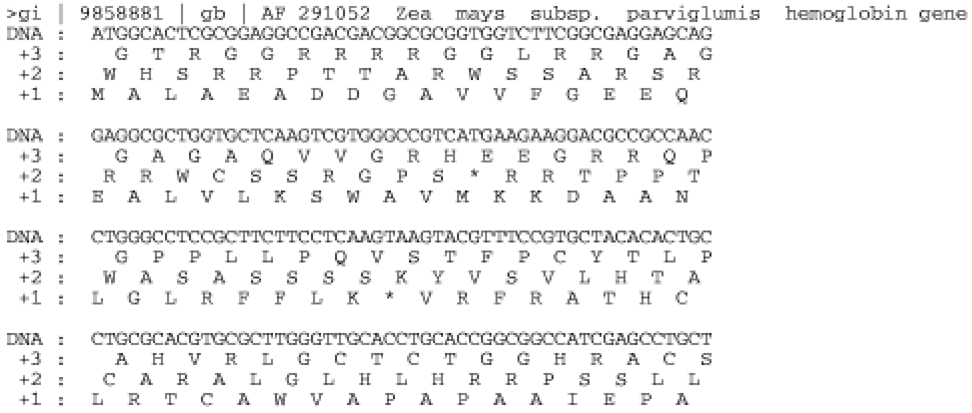
- Knowledge-based single sequence analysis for sequence characteristics
- Pairwise sequence comparison and sequence-based searching
- Multiple sequence alignment
- Sequence motif discovery in multiple alignments
- Phylogenetic inference

We divide our coverage of sequence analysis tools into two chapters. This chapter focuses on programs that operate on single sequences, or compare gene or protein sequences against each other. Chapter 8 is devoted to multiple sequence alignment methods.

Pairwise sequence comparison is the primary means of linking biological function to the genome and of propagating known information from one genome to another. In this chapter, we discuss the techniques of biological sequence analysis and, most importantly, how to assess the significance of results from sequence comparison. There are also a number of software tools available for doing pairwise sequence comparison. Table 7-1 provides a summary.

Table 7-1. Sequence Analysis Tools and Techniques

What you do	Why you do it	What you use to do it
Gene finding	Identify possible coding regions in genomic DNA sequences	GENSCAN, GeneWise, PROCRUSTES, GRAIL
DNA feature detection	Locate splice sites, promoters, and sequences involved in regulation of gene expression	CBS Prediction Server
DNA translation and reverse translation	Convert a DNA sequence into protein sequence or vice versa	"Protein machine" server at EBI
Pairwise sequence alignment (local)	Locate short regions of homology in a pair of longer sequences	BLAST, FASTA
Pairwise sequence alignment (global)	Find the best full-length alignment between two sequences	ALIGN
Sequence database search by pairwise comparison	Find sequence matches that aren't recognized by a keyword search; find only matches that actually have some sequence homology	BLAST, FASTA, SSEARCH



```
>gi | 9858881 | gb | AF 291052  Zea  mays  subsp.  parviglumis  hemoglobin gene
DNA :  ATGGCACTCGCGGAGGCCGACGACGGCGCGGTGGTCTTCGGCGAGGAGCAG
 +3 :    G  T  R  G  G  R  R  R  R  G  G  L  R  R  G  A  G
 +2 :   W  H  S  R  R  P  T  T  A  R  W  S  S  A  R  S  R
 +1 :  M  A  L  A  E  A  D  D  G  A  V  V  F  G  E  E  Q

DNA :  GAGGCGCTGGTGCTCAAGTCGTGGGCCGTCATGAAGAAGGACGCCGCCAAC
 +3 :    G  A  G  A  Q  V  V  G  R  H  E  E  G  R  R  Q  P
 +2 :   R  R  W  C  S  S  R  G  P  S  *  R  R  T  P  P  T
 +1 :  E  A  L  V  L  K  S  W  A  V  M  K  K  D  A  A  N

DNA :  CTGGGCCTCCGCTTCTTCCTCAAGTAAGTACGTTTCCGTGCTACACACTGC
 +3 :    G  P  P  L  L  P  Q  V  S  T  F  P  C  Y  T  L  P
 +2 :   W  A  S  A  S  S  S  S  K  Y  V  S  V  L  H  T  A
 +1 :  L  G  L  R  F  F  L  K  *  V  R  F  R  A  T  H  C

DNA :  CTGCGCACGTGCGCTTGGGTTGCACCTGCACCGGCGGCCATCGAGCCTGCT
 +3 :    A  H  V  R  L  G  C  T  C  T  G  G  H  R  A  C  S
 +2 :   C  A  R  A  L  G  L  H  L  H  R  R  P  S  S  L  L
 +1 :  L  R  T  C  A  W  V  A  P  A  P  A  A  I  E  P  A
```

Because of the large number of codon possibilities for some amino acids, back-translation of a protein into DNA sequence can result in an extremely large number of possible sequences. However, codon usage statistics for different species are available and can be used to suggest the most likely back-translation out of the range of possibilities.

BLAST and FASTA dynamically translate query and database sequences so you don't need to worry about translating a database before you do a sequence comparison. However, in the event that you need to produce a six-frame translation of a single DNA sequence or translate a protein back into a set of possible DNA sequences, and you don't want to script it yourself, the Protein Machine server (http://www.ebi.ac.uk/translate/) at the European Bioinformatics Institute (EBI) will do it for you.

7.7 Pairwise Sequence Comparison

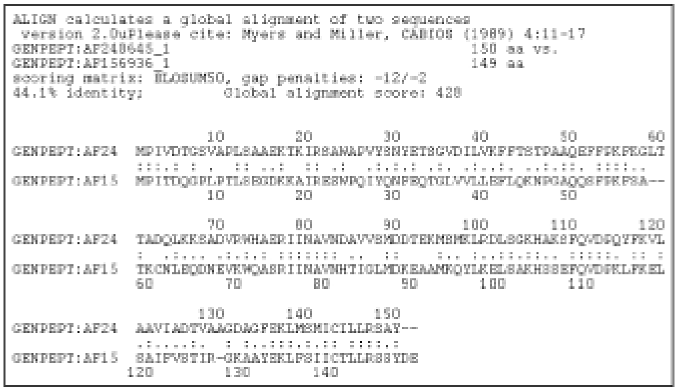
Comparison of protein and DNA sequences is one of the foundations of bioinformatics. Our ability to perform rapid automated comparisons of sequences facilitates everything from assignment of function to a new sequence, to prediction and construction of model protein structures, to design and analysis of gene expression experiments. As biological sequence data has accumulated, it has become apparent that nature is conservative. A new biochemistry isn't created for each new species, and new functionality isn't created by the sudden appearance of whole new genes. Instead, incremental modifications give rise to genetic diversity and novel function. With this premise in mind, detection of similarity between sequences allows you to transfer information about one sequence to other similar sequences with reasonable, though not always total, confidence.

Before you can make comparative statements about nucleic acid or protein sequences, a sequence alignment is needed. The basic concept of selecting an optimal sequence alignment is simple. The two sequences are matched up in an arbitrary way. The quality of the match is scored. Then one sequence is moved with respect to the other and the match is scored again, until the best-scoring alignment is found.

What sounds simple in principle isn't at all simple in practice. Choosing a good alignment by eye is possible, but life is too short to do it more than once or twice. An automated method for finding the optimal alignment out of the thousands of alternatives is clearly the right approach, but in order for the method to be consistent and biologically meaningful, several questions must be answered. How should alignments be scored? A scoring scheme can be as simple as +1 for a match and -1 for a mismatch, but what is the best scoring scheme for the data? Should gaps be allowed to open in the sequences to facilitate better matches elsewhere? If gaps are allowed, how should they be scored? Given the correct scoring parameters, what is the best algorithm for finding the optimal alignment of two sequences? And when an alignment is produced, is it necessarily significant? Can an alignment of similar quality be produced for two random sequences? Through the rest of this section, we consider each of these questions in greater detail.

Figure 7-8 shows examples of three kinds of alignment. These are three pairwise sequence alignments generated using a program called ALIGN. In each alignment, the sequences being compared are displayed, one above the other, such that matching residues are aligned. Identical matches are indicated with a colon (:) between the matching residues, while similarities are indicated with a single dot (.). Information about the alignment is presented at the top, including percent identity (the number of identical matches divided by the length of the alignment) and score. Finally, gaps in one sequence relative to another are represented by dashes (-) for each position in that sequence occupied by a gap.

Figure 7-8. Three alignments: high scoring, low scoring but meaningful, and random

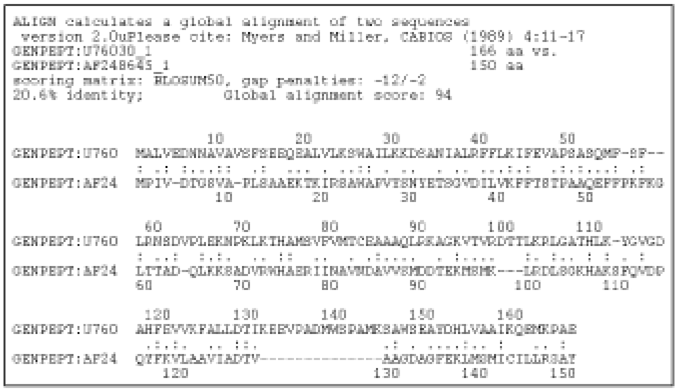


```
ALIGN calculates a global alignment of two sequences
 version 2.0uPlease cite: Myers and Miller, CABIOS (1989) 4:11-17
GENPEPT:AF248645_1                                150 aa vs.
GENPEPT:AF156936_1                                149 aa
scoring matrix: BLOSUM50, gap penalties: -12/-2
44.1% identity;         Global alignment score: 428


                       10        20        30        40        50        60
GENPEPT:AF24   MPIVDTGSVAPLSAAEKTKIRSAWAPVYSNYETSGVDILVKFFTSTPAAQEFFPKFKGLT
               :::.: : .  :: ..:  :: .:  .:.:.: .:. .:..:. ..:.::. ::::..
GENPEPT:AF15   MPITDQGPLPTLSEGDKKAIRESWPQIYQNFEQTGLVVLLEFLQKNPGAQQSFPKFSA--
                       10        20        30        40        50

                       70        80        90       100       110       120
GENPEPT:AF24   TADQLKKSADVRWHAERIINAVNDAVVSMDDTEKMSMKLRDLSGKHAKSFQVDPQYFKVL
               :  .:... .:.:.: :::::: ..  ::    :.. :..::.::.. :::::. :: :
GENPEPT:AF15   TKCNLEQDNEVKWQASRIINAVNHTIGLMDKEAAMKQYLKELSAKHSSEFQVDPKLFKEL
                60        70        80        90       100       110

                      130       140       150
GENPEPT:AF24   AAVIADTVAAGDAGFEKLMSMICILLRSAY--
               .:....:.  : :..:::.:.:: ::::.:
GENPEPT:AF15   SAIFVSTIR-GKAAYEKLFSIICTLLRSSYDE
              120       130       140
```

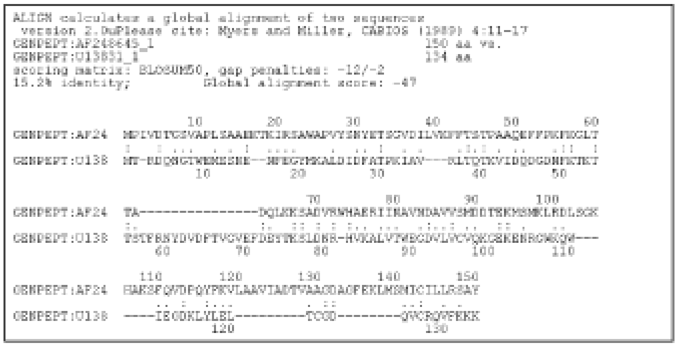


```
ALIGN calculates a global alignment of two sequences
 version 2.0uPlease cite: Myers and Miller, CABIOS (1989) 4:11-17
GENPEPT:U76030_1                                  166 aa vs.
GENPEPT:AF248645_1                                150 aa
scoring matrix: BLOSUM50, gap penalties: -12/-2
20.6% identity;         Global alignment score: 94


                       10        20        30        40        50
GENPEPT:U760   MALVEDNNAVAVSFSEEQEALVLKSWAILKKDSANIALRFFLKIFEVAPSASQMF-SF--
               : .: :...::  .:   ... . ..:: . ..   . .. ...:.:  .:.:..: .:
GENPEPT:AF24   MPIV-DTGSVA-PLSAAEKTKIRSAWAPVYSNYETSGVDILVKFFTSTPAAQEFFPKFKG
                        10         20        30        40        50

                 60        70        80        90       100       110
GENPEPT:U760   LRNSDVPLEKNPKLKTHAMSVFVMTCEAAAQLRKAGKVTVRDTTLKRLGATHLK-YGVGD
               : ..:  :.:.  .. ::  ..   . .:....  . :....    :. :.. : : . :
GENPEPT:AF24   LTTAD-QLKKSADVRWHAERIINAVNDAVVSMDDTEKMSMK---LRDLSGKHAKSFQVDP
                60        70        80        90          100       110

                  120       130       140       150       160
GENPEPT:U760   AHFEVVKFALLDTIKEEVPADMWSPAMKSAWSEAYDHLVAAIKQEMKPAE
                .:.:.  .. ::.                .: . ....:.. :   .. :
GENPEPT:AF24   QYFKVLAAVIADTV--------------AAGDAGFEKLMSMICILLRSAY
                    120                     130       140       150
```



```
ALIGN calculates a global alignment of two sequences
 version 2.0uPlease cite: Myers and Miller, CABIOS (1989) 4:11-17
GENPEPT:AF248645_1                                150 aa vs.
GENPEPT:U13831_1                                  134 aa
scoring matrix: BLOSUM50, gap penalties: -12/-2
15.2% identity;         Global alignment score: -47


                       10        20        30        40        50        60
GENPEPT:AF24   MPIVDTGSVAPLSAAEKTKIRSAWAPVYSNYETSGVDILVKFFTSTPAAQEFFPKFKGLT
               :  : ... . . :  ....     .  .. :  . .    .:.: . ..   .:: :
GENPEPT:U138   MT-RDQNGTWEMESNE--NFEGYMKALDIDFATRKIAV---RLTQTKVIDQDGDNFKTKT
                        10         20        30           40        50

                                          70        80        90       100
GENPEPT:AF24   TA---------------DQLKKSADVRWHAERIINAVNDAVVSMDDTEKMSMKLRDLSGK
               :.               :.  :: : : :.. ...  .:..: ..  ::: .   ..
GENPEPT:U138   TSTFRNYDVDFTVGVEFDEYTKSLDNR-HVKALVTWEGDVLVCVQKGEKENRGWKQW---
                    60        70        80         90       100       110

                     110       120       130       140       150
GENPEPT:AF24   HAKSFQVDPQYFKVLAAVIADTVAAGDAGFEKLMSMICILLRSAY
                   .. :  :...         . ::        ..:  . .
GENPEPT:U138   ----IEGDKLYLEL---------TCGD--------QVCRQVFKKK
                            120                       130
```

The first alignment is a high-scoring one: it shows a comparison of two closely related proteins (two hemoglobin molecules, one from a sea lamprey and one from a hagfish). Compare that alignment with the second, a comparison of two distantly related proteins (again, two hemoglobin molecules, in this case taken from lamprey and rice). Cursory inspection shows fewer identical residues are shared by the sequences in the low-scoring alignment than in the high-scoring one. Still, there are several similarities or conservative changes—changes in which one amino acid has been replaced by another, chemically similar residue. The third alignment is a random alignment, a comparison between two unrelated sequences (the lamprey hemoglobin and a human retinol binding protein). Notice that, in addition to the few identities and conservative mutations between the two, large gaps have been opened in both sequences to achieve this alignment. Gene families aren't likely to evolve in this way, and given the lack of similarity between the sequences, you can conclude that these proteins are unrelated.

In describing sequence comparisons, several different terms are commonly used. Sequence identity, sequence similarity, and sequence homology are the most important of these terms. Each means something slightly different, though they are often casually used interchangeably.

Sequence identity refers to the occurrence of exactly the same nucleic acid or amino acid in the same position in two aligned sequences. *Sequence similarity* is meaningful only when possible substitutions are scored according to the probability with which they occur. In protein sequences, amino acids of similar chemical properties are found to substitute for each other much more readily than dissimilar amino acids. These propensities are represented in scoring matrices that score sequence alignments. Two amino acids are considered similar if one can be substituted for another with a positive log odds score from a scoring matrix (described in the next section).

Sequence homology is a more general term that indicates evolutionary relatedness among sequences. It is common to speak of a percentage of sequence homology when comparing two sequences, although that percentage may indicate a mixture of identical and similar sites. Finally, sequence homology refers to the evolutionary relatedness between sequences. Two sequences are said to be homologous if they are both derived from a common ancestral sequence. The terms similarity and homology are often used interchangeably to describe sequences, but, strictly speaking, they mean different things. Similarity refers to the presence of identical and similar sites in the two sequences, while homology reflects a stronger claim that the two sequences share a common ancestor.

7.7.1 Scoring Matrices

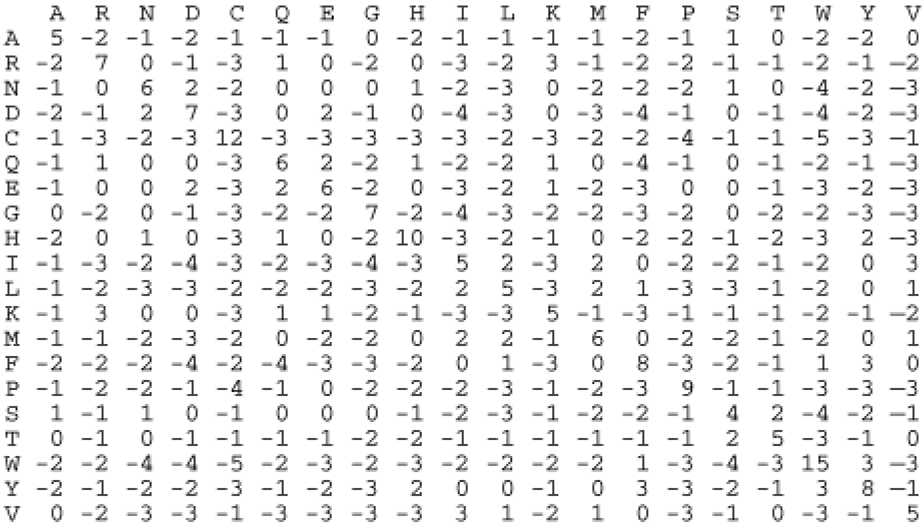
What you really want to learn when evaluating a sequence alignment is whether a given alignment is random, or meaningful. If the alignment is meaningful, you want to gauge just how meaningful it is. You attempt to do this by constructing a scoring matrix.

A scoring matrix is a table of values that describe the probability of a residue (amino acid or base) pair occurring in an alignment. The values in a scoring matrix are logarithms of ratios of two probabilities. One is the probability of random occurrence of an amino acid in a sequence alignment. This value is simply the product of the independent frequencies of occurrence of each of the amino acids. The other is the probability of meaningful occurrence of a pair of residues in a sequence alignment. These probabilities are derived from samples of actual sequence alignments that are known to be valid.

In order to score an alignment, the alignment program needs to know if it is more likely that a given amino acid pair has occurred randomly, or that it has occurred as a result of an evolutionary event. The logarithm of the ratio of the probability of meaningful occurrence to the probability of random occurrence is positive if the probability of meaningful occurrence is greater, and negative if the probability of random occurrence is greater. Because the scores are logarithms of probability ratios, they can be meaningfully added to give a score for the entire sequence. The more positive the score, the more likely the alignment is to be significant.

Figure 7-9 shows an example of a BLOSUM45 matrix, a popular substitution matrix for amino acids.

Figure 7-9. The BLOSUM45 matrix, a popular substitution matrix for amino acids



	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-2	-2	0
R	-2	7	0	-1	-3	1	0	-2	0	-3	-2	3	-1	-2	-2	-1	-1	-2	-1	-2
N	-1	0	6	2	-2	0	0	0	1	-2	-3	0	-2	-2	-2	1	0	-4	-2	-3
D	-2	-1	2	7	-3	0	2	-1	0	-4	-3	0	-3	-4	-1	0	-1	-4	-2	-3
C	-1	-3	-2	-3	12	-3	-3	-3	-3	-3	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1
Q	-1	1	0	0	-3	6	2	-2	1	-2	-2	1	0	-4	-1	0	-1	-2	-1	-3
E	-1	0	0	2	-3	2	6	-2	0	-3	-2	1	-2	-3	0	0	-1	-3	-2	-3
G	0	-2	0	-1	-3	-2	-2	7	-2	-4	-3	-2	-2	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	0	-3	1	0	-2	10	-3	-2	-1	0	-2	-2	-1	-2	-3	2	-3
I	-1	-3	-2	-4	-3	-2	-3	-4	-3	5	2	-3	2	0	-2	-2	-1	-2	0	3
L	-1	-2	-3	-3	-2	-2	-2	-3	-2	2	5	-3	2	1	-3	-3	-1	-2	0	1
K	-1	3	0	0	-3	1	1	-2	-1	-3	-3	5	-1	-3	-1	-1	-1	-2	-1	-2
M	-1	-1	-2	-3	-2	0	-2	-2	0	2	2	-1	6	0	-2	-2	-1	-2	0	1
F	-2	-2	-2	-4	-2	-4	-3	-3	-2	0	1	-3	0	8	-3	-2	-1	1	3	0
P	-1	-2	-2	-1	-4	-1	0	-2	-2	-2	-3	-1	-2	-3	9	-1	-1	-3	-3	-3
S	1	-1	1	0	-1	0	0	0	-1	-2	-3	-1	-2	-2	-1	4	2	-4	-2	-1
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-1	-1	2	5	-3	-1	0
W	-2	-2	-4	-4	-5	-2	-3	-2	-3	-2	-2	-2	-2	1	-3	-4	-3	15	3	-3
Y	-2	-1	-2	-2	-3	-1	-2	-3	2	0	0	-1	0	3	-3	-2	-1	3	8	-1
V	0	-2	-3	-3	-1	-3	-3	-3	-3	3	1	-2	1	0	-3	-1	0	-3	-1	5

Substitution matrices for amino acids are complicated because they reflect the chemical nature and frequency of occurrence of the amino acids. For example, in the BLOSUM matrix, glutamic acid (E) has a positive score for substitution with aspartic acid (D) and also with glutamine (Q). Both these substitutions are chemically conservative. Aspartic acid has a sidechain that is chemically similar to glutamic acid, though one methyl group shorter. On the other hand, glutamine is similar in size and chemistry to glutamic acid, but it is neutral while glutamic acid is negatively charged. Substitution scores for glutamic acid with residues such as isoleucine (I) and leucine (L) are negative. These residues have neutral, nonpolar sidechains and are chemically different from glutamic acid. The scores on the diagonal of the matrix reflect the frequency of occurrence of each amino acid. For example, with a positive score of 15, it is extremely unlikely that the alignment of a rare tryptophan (W) with another tryptophan is coincidence, while the more common serine (S) has a positive score of only 4 for a match with another serine. It's important to remember that these scores are logarithms, which means that a match of two serines is far from being mere coincidence.

Substitution matrices for bases in DNA or RNA sequence are very simple. By default, the sequence alignment program BLAST uses the scheme of assigning a standard reward for a match and a standard penalty for a mismatch, with no regard to overall frequencies of bases. In most cases, it is reasonable to assume that A:T and G:C occur in roughly equal proportions.

Commonly used substitution matrices include the BLOSUM and PAM matrices. When using BLAST, you need to select a scoring matrix. Most automated servers select a default matrix for you (usually something like BLOSUM62), and if you're just doing a quick sequence search, it's fine to accept the default.

BLOSUM matrices are derived from the Blocks database, a set of ungapped alignments of sequence regions from families of related proteins. A clustering approach sorts the sequences in each block into closely related groups, and the frequencies of substitutions between these within a family derives the probability of a meaningful substitution. The numerical value (e.g., 62) associated with a BLOSUM matrix represents the cutoff value for the clustering step. A value of 62 indicates that sequences were put into the same cluster if they were more than 62% identical. By allowing more diverse sequences to be included in each cluster, lower cutoff values represent longer evolutionary time scales, so matrices with low cutoff values are appropriate for seeking more distant relationships. BLOSUM62 is the standard matrix for ungapped alignments, while BLOSUM50 is more commonly used when generating alignments with gaps.

Point accepted mutation (PAM) matrices are scaled according to a model of evolutionary distance from alignments of closely related sequences. One PAM "unit" is equivalent to an average change in 1% of all amino acid positions. The most commonly used PAM matrix is PAM250. However, comparison of results using PAM and BLOSUM matrices suggest that BLOSUM matrices are better at detecting biologically significant similarities.

7.7.2 Gap Penalties

DNA sequences change not only by point mutation, but by insertion and deletion of residues as well. Consequently, it is often necessary to introduce gaps into one or both of the sequences being aligned to produce a meaningful alignment between them. Most algorithms use a gap penalty to represent the validity of adding a gap in an alignment.

The addition of a gap has to be costly enough, in terms of the overall alignment score, that gaps will open only where they are really needed and not all over the sequence. Most sequence alignment models use affine gap penalties, in which the cost of opening a gap in a sequence is different from the cost of extending a gap that has already been started. Of these two penalties—-the gap opening penalty and the gap extension penalty—-the gap opening penalties tend to be much higher than the associated extension penalty. This tendency reflects the tendency for insertions and deletions to occur over several residues at a time.

Gap penalties are intimately tied to the scoring matrix that aligns the sequences: the best pair of gap opening and extension penalties for one scoring matrix doesn't necessarily work with another. Scores of -11 for gap opening and -1 for gap extension are commonly used in conjunction with the BLOSUM 62 matrix for gapped-BLAST, while BLOSUM50 uses a -12/-1 penalty.

7.7.3 Dynamic Programming

Dynamic programming methods are a general class of algorithms that are often seen both in sequence alignment and other computational problems. They were first described in the 1950s by Richard Bellman of Princeton University as a general optimization technique. Dynamic programming seems to have been introduced[2] to biological sequence comparison by Saul Needleman and Christian Wunsch, who apparently were unaware of the similarity between their method and Bellman's.

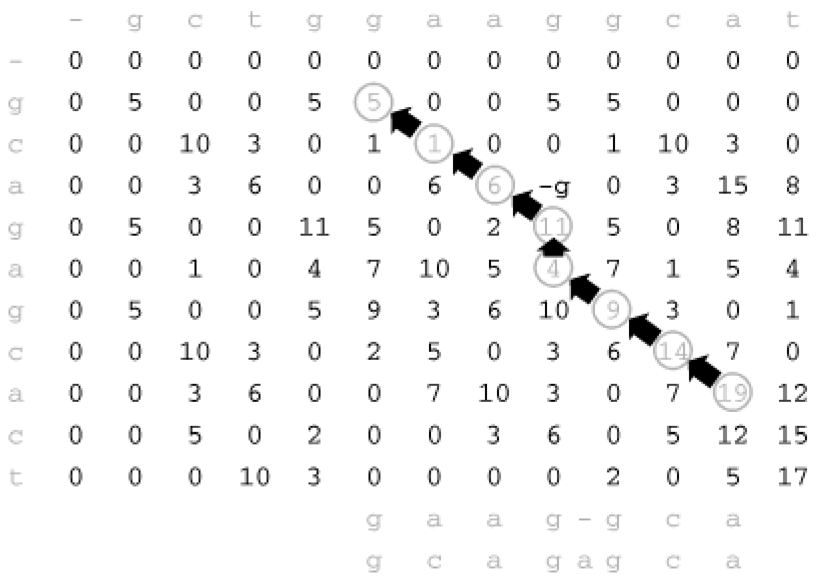
As we mentioned, dynamic programming algorithms solve optimization problems, problems in which there are a large number of possible solutions, but only one (or a small number of) best solutions. A dynamic programming algorithm finds the best solution by first breaking the original problem into smaller subproblems and then solving. These pieces of the larger problem have a sequential dependency; that is, the fourth piece can be solved only with the answer to the third piece, the third can be solved only with the answer to the second, and so on. Dynamic programming works by first solving all these subproblems, storing each intermediate solution in a table along with a score, and finally choosing the sequence of solutions that yields the highest score. The goal of the dynamic programming algorithm is to maximize the total score for the alignment. In order to do this, the number of high-scoring residue pairs must be maximized and the number of gaps and low-scoring pairs must be minimized.

In sequence comparison, the overall problem is finding the best alignment between two sequences. This problem is broken down into subproblems of aligning each residue from one sequence with each residue from the other. The solution is a decision as to whether the residues should be aligned with each other, a gap should be introduced in the first sequence, or a gap should be introduced in the second sequence. Each high-scoring choice rules out the other two low-scoring possibilities, so that if information about the accumulated scores is stored at each step, every possible alignment need not be evaluated.

The algorithm uses an (*m* x *n*) matrix of scores (illustrated in Figure 7-10) in which *m* and *n* are the lengths of the sequences being aligned. Starting with the alignment of a gap against itself (which is given the initial score zero), the algorithm fills in the matrix one row at a time. At each position in the matrix, the algorithm computes the scores that result for each of its three choices, selects the one that yields the highest score, then stores a pointer at the current position to the preceding position that was used to arrive at the high score. When every position in the matrix has been filled in, a traceback step is performed, and the highest-scoring path along the pointers is followed back to the beginning of the alignment.

Figure 7-10. A matrix of scores comparing two sequences; continuous high-scoring matches are highlighted

[2] Or, as mathematicians might say, "rediscovered." Because computational biology combines research from so many different areas, this independent discovery happens often and is only noticed later.



	-	g	c	t	g	g	a	a	g	g	c	a	t
-	0	0	0	0	0	0	0	0	0	0	0	0	0
g	0	5	0	0	5	5	0	0	5	5	0	0	0
c	0	0	10	3	0	1	1	0	0	1	10	3	0
a	0	0	3	6	0	0	6	6	-g	0	3	15	8
g	0	5	0	0	11	5	0	2	11	5	0	8	11
a	0	0	1	0	4	7	10	5	4	7	1	5	4
g	0	5	0	0	5	9	3	6	10	9	3	0	1
c	0	0	10	3	0	2	5	0	3	6	14	7	0
a	0	0	3	6	0	0	7	10	3	0	7	19	12
c	0	0	5	0	2	0	0	3	6	0	5	12	15
t	0	0	0	10	3	0	0	0	0	2	0	5	17
						g	a	a	g -	g	c	a	
						g	c	a	g a	g	c	a	

7.7.4 Global Alignment

One alignment scenario you may encounter is the alignment of two sequences along their whole length. The algorithm for alignment of whole sequences is called the Needleman-Wunsch algorithm. In this scenario, an optimal alignment is built up from high-scoring alignments of subsequences, stepping through the matrix from top left to bottom right. Only the best-scoring path can be traced through the matrix, resulting in an optimal alignment.

7.7.4.1 Using ALIGN to produce a global sequence alignment

Now that we have seen the theory behind the global alignment of two sequences, let's examine a program that implements this algorithm. ALIGN is a simple utility for computing global alignments. It is part of the FASTA software distribution, described later in this chapter. The programs in the FASTA distribution are easily run from the Unix command line, although many of them have been incorporated into the SDSC Biology Workbench web-based sequence analysis software, if you prefer to access them through a point-and-click interface. The FASTA programs compile easily under Linux; however, once they are compiled, you need to link them into your */usr/local/bin* directory or some other sensible location by hand.

To run ALIGN and any of the other FASTA programs, you need sequence data in FASTA format. This is one of the most frequently used sequence formats and probably the simplest. To use ALIGN, each of the sequences you are aligning should be in a separate file.

A sequence in FASTA format[3] looks like this:

[3] Also known as Pearson format, after the author of the FASTA software, William Pearson.

```
>2HHB:A HEMOGLOBIN (DEOXY) - CHAIN A
VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGK
KVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPA
```

```
VHASLDKFLASVSTVLTSKYR
```

The FASTA format is very flexible, and it is one of the most commonly used formats for sequence analysis programs. A FASTA file contains one or more records in FASTA format, separated by empty lines. Each record consists of a human-readable comment followed by a nucleotide or protein sequence. The comment appears in the first line of the record; it must begin with a greater-than (>) symbol followed by one or more identifiers for the sequence. The comment may contain information about the molecule represented by the sequence, such as the protein or gene name and source organism. In the previous example, the identifier is a PDB code (2HHB), followed by a description of the sequence (the A chain of a deoxyhemoglobin protein). The remainder of the record contains the sequence itself, divided into separate lines by line breaks. Lines are usually 60 characters long, but the format doesn't specify a line length. Programs that take FASTA data as input (such as ALIGN) usually make allowances for FASTA's free-form nature. Still, it's a good practice to check the program's documentation to make sure that your data is appropriately formatted.

To use ALIGN, simply enter *align* at the command prompt. You are then prompted for sequence filenames. Results are sent to standard output. The ASCII format for pairwise alignments that is produced by FASTA is still commonly used, although there is a trend toward use of more easily parsed alignment formats:

```
Output scoring matrix: BLOSUM50, gap penalties: -12/-2 43.2% identity;
Global alignment score: 374

                10        20        30        40              50
2HHB_A V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSA
       : :.: .:. : : ::::  .. : :.::: :... .: :. .:  : :::      :.
2HHB:B VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP

               10          20        30        40        50
            60        70        80        90       100       110
2HHB_A QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL
       .::.:::::  :.....::.:.. .....::.::  ::.::: ::.::.. :. .:: :.
2HHB:B KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF
       60        70        80        90       100       110
           120       130       140
2HHB_A PAEFTPAVHASLDKFLASVSTVLTSKYR
         :::: :.:. .: .:.:...:. ::.
2HHB:B GKEFTPPVQAAYQKVVAGVANALAHKYH
      120       130       140
```

The FASTA distribution contains a sample HTML form and CGI script for use with the program LALIGN, a pairwise local alignment program. The script can be modified to work with the ALIGN program if a web-based interface is desired.

7.7.5 Local Alignment

The most commonly used sequence alignment tools rely on a strategy called *local alignment*. The global alignment strategy discussed earlier assumes that the two sequences to be aligned are known and are to be aligned over their full length. In the scenarios that are encountered most often with sequence alignment, however, you are either searching with one sequence against a sequence database looking for unknown sequences, or searching a very long DNA sequence, such as part of a genome, for partial segments that match a query sequence. In protein or gene sequences that do have some evolutionary relatedness, but which have diverged significantly from each other, short homologous segments may be all the evidence of sequence homology that remains.

The version of the dynamic programming algorithm that performs local alignment of two sequences is known as the Smith-Waterman algorithm. Named for its inventors, Dr. Temple Smith and Dr. Michael Waterman, this algorithm is similar to the Needleman-Wunsch algorithm except that an additional choice is allowed when tracing through the matrix. A local alignment isn't required to extend from beginning to end of the two sequences being aligned. If the cumulative score up to some point in the sequence is negative, the alignment can be abandoned and a new alignment started. The alignment can also end anywhere in the matrix.

7.7.5.1 Tools for local alignment

One of the most frequently reported implementations of the Smith-Waterman algorithm for database searching is the program SSEARCH, which is part of the FASTA distribution described later. LALIGN, also part of the FASTA package, is an implementation of the Smith-Waterman algorithm for aligning two sequences.

7.8 Sequence Queries Against Biological Databases

A common application of sequence alignment is searching a database for sequences that are similar to a query sequence. In these searches, an alignment of a sequence hundreds or thousands of residues long is matched against a database of at least tens of thousands of comparably sized sequences. Using dynamic programming-based methods, this isn't very practical unless special-purpose hardware is available. Instead, for routine searches, special heuristic database-searching methods are used. Heuristic methods exploit knowledge about sequences and alignment statistics to make these large-scale searches efficient and practical. While they don't guarantee optimal alignments, they make sensitive searches of large sequence databases possible. In this section, we describe BLAST and FASTA, two commonly used database-searching programs.

7.8.1 Local Alignment-Based Searching Using BLAST

By far, the most popular tool for searching sequence databases is a program called BLAST (Basic Local Alignment Search Tool). BLAST is the algorithm at the core of most online sequence search servers.[4] It performs pairwise comparisons of sequences, seeking regions of local similarity, rather than optimal global alignments between whole sequences.

[4] To give you perspective on how long the common tools of bioinformatics have been available, the original BLAST paper by Altschul et al. was published in the *Journal of Molecular Biology* in October 1990.

BLAST can perform hundreds or even thousands of sequence comparisons in a matter of minutes. And in less than a few hours, a query sequence can be compared to an entire database to find all similar sequences. BLAST is so popular for this purpose that it's become a verb in the computational biology community, as in "I BLASTed this sequence against GenBank and came up with three matches."

7.8.1.1 The BLAST algorithm

Local sequence alignment searching using a standard Smith-Waterman algorithm is a fairly slow process. The BLAST algorithm, which speeds up local sequence alignment, has three basic steps. First, it creates a list of all short sequences (called *words* in BLAST terminology) that score above a threshold value when aligned with the query sequence. Next, the sequence database is searched for occurrences of these words. Because the word length is so short (3 residues for proteins, 11 residues for nucleic acids), it's possible to search a precomputed table of all words and their positions in the sequences for improved speed. These matching words are then extended into ungapped local alignments between the query sequence and the sequence from the database. Extensions are continued until the score of the alignment drops below a threshold. The top-scoring alignments in a sequence, or maximal-scoring segment pairs (MSPs), are combined where possible into local alignments. Originally, BLAST searched only for ungapped alignments. However, new additions to the BLAST software package that do search for gapped alignments have since been introduced.

7.8.1.2 NCBI BLAST and WU-BLAST

There are two implementations of the BLAST algorithm: NCBI BLAST and WU-BLAST. Both implementations can be used as web services and as downloadable software packages. NCBI BLAST is available from the National Center for Biotechnology Information (NCBI), while WU-BLAST is an alternate version that grew out of NCBI BLAST 1.4 and is developed and maintained by Dr. Warren Gish and coworkers at Washington University.

NCBI BLAST is the more commonly used of the two. The most recent versions of this program have focused on the development of methods for comparing multiple-sequence profiles (see Chapter 8). WU-BLAST, on the other hand, has developed a different system for handling gaps as well as a number of features (such as filtering for repeats) that are useful for searching genome sequences. Consequently, TIGR, Stanford's yeast genome server, Berkeley's Drosophila genome project, and others use WU-BLAST 2.0 as the sequence-comparison tool for searching their genome sequence data via the Web. As of this writing, WU-BLAST 2.0, the most recent version of the software, is copyrighted. NCBI BLAST and WU-BLAST's previous version, 1.4, are both in the public domain and freely available to all researchers. Because of its ubiquity we focus on NCBI BLAST in the following section, but WU-BLAST is an alternative. For more information on these flavors of BLAST see the NCBI web site at http://www.ncbi.nlm.nih.gov/BLAST, or the WU-BLAST web site at http://blast.wustl.edu.

7.8.1.3 What do the various BLAST programs do?

Frequent users of BLAST can also download and install BLAST binaries on their own machines. BLAST installs easily on a Linux system. Simply create a new directory (e.g., */usr/local/blast*), move the archive into it, and extract. Here are the four main executable programs in the BLAST distribution:

[blastall]

Performs BLAST searches using one of five BLAST programs: *blastp, blastn, blastx, tblastn,* or *tblastx*

[blastpgp]

Performs searches in PSI-BLAST or PHI-BLAST mode

[bl2seq]

Performs a local alignment of two sequences

[formatdb]

Converts a FASTA-format flat file sequence database into a BLAST database

blastall encompasses all the major options for ungapped and gapped BLAST searches. A full list of its command-line arguments can be displayed with the command *blastall* - :

[-p]

Program name. Its options include:

blastp

Protein sequence (PS) query versus PS database

blastn

Nucleic acid sequence (NS) query versus NS database

blastx

NS query translated in all six reading frames versus PS database

tblastn

PS query versus NS database dynamically translated in all six reading frames

tblastx

Translated NS query versus translated NS database—computationally intensive

[-d]

Database name. Each indexed database consists of several files; the name is the common portion of those filenames.

The Biology Workbench offers keyword and sequence-based searching of nearly 40 major sequence databases and over 25 whole genomes. Both BLAST and FASTA are implemented as search and alignment tools in the Workbench, along with several local and global alignment tools, tools for DNA sequence translation, protein sequence feature analysis, multiple sequence alignment, and phylogenetic tree drawing. The Workbench group has not yet implemented profile tools, such as MEME, HMMer, or sequence logo tools, although PSI-BLAST is available for sequence searches.

Although its interface can be somewhat cumbersome, involving a lot of window scrolling and button clicking, the Biology Workbench is still the most comprehensive, convenient, and accessible of the web-based toolkits. One of its main benefits is that many sequence file formats are accepted and translated by the software. Users of the Workbench need never worry about file type incompatibility and can move seamlessly from keyword-based database search, to sequence-based search, to multiple alignment, to phylogenetic analysis.

7.9.3 DoubleTwist

Another entry into the sequence analysis portal arena is DoubleTwist at http://doubletwist.com. This site allows you to submit a sequence for comparison to multiple databases using BLAST. It also provides "agents" that monitor databases for new additions that match a submitted sequence and automatically notifies the user. These services, as well as access to the EcoCyc pathways database and to an online biology research protocols database, are free with registration at the site at the time of this writing.

Chapter 8. Multiple Sequence Alignments, Trees, and Profiles

In Chapter 7, we introduced the idea of using sequence alignment to find and compare pairs of related sequences. Biologically interesting problems, however, often involve comparing more than two sequences at once. For example, a BLAST or FASTA search can yield a large number of sequences that match the query. How do you compare all these resulting sequences with each other? In other words, how can you examine these sequences to understand how they are related to one another?

One approach is to perform pairwise alignments of all pairs of sequences, then study these pairwise alignments individually. It's more efficient (and easier to comprehend), however, if you compare all the sequences at once, then examine the resulting ensemble alignment. This process is known as *multiple sequence alignment*. Multiple sequence alignments can be used to study groups of related genes or proteins, to infer evolutionary relationships between genes, and to discover patterns that are shared among groups of functionally or structurally related sequences. In this chapter, we introduce some tools for creating and interpreting multiple sequence alignments and describe some of their applications, including phylogenetic inference and motif discovery. Phylogenetic inference and motif discovery are rooted in evolutionary theory, so before we dive into a discussion of that area of bioinformatics, let's take a minute to review the history and theory of evolution.

8.1 The Morphological to the Molecular

In order to ground our discussion of the details of multiple sequence alignment, let's take another brief look at evolution. One of the goals of biology has been the creation of a taxonomy for living things, a method of organizing species in terms of their relationships to one another. Early biologists classified species solely according to their morphology—the physical appearance of the organism—and later, as dissection became a more common practice, their anatomy.

Naturalists also discovered fossils of creatures that didn't resemble anything alive at the time, but were thought to have once been living things. This evidence introduced the possibility that life on Earth had changed over time. It also suggested that the interrelationship between species isn't static, but rather is the result of an evolutionary process. As understanding of the geophysical processes of the planet improved, the amount of time required for such changes to occur became clearer. It is now widely accepted by scientists that life on Earth is approximately 3.5 billion years old. Fossil records of single-celled organisms resembling bacteria, with an estimated age of 3.5 billion years, have been found and catalogued.

The evolutionary theory that was eventually accepted by most biologists was that of Charles Darwin. Darwin proposed that every generation of living creatures has some variability. The individuals whose variations predispose them to survive in their environment are the ones who reproduce most successfully and pass on their traits in greater numbers. In light of this theory, it has been hypothesized that the diversity of life forms on Earth is due to divergence, perhaps even from one common ancestral unicellular organism, to fill various biological niches.

Molecular evolution extends the concept of evolution to the level of DNA and protein sequences. Although the replication of DNA sequence is a very accurate process, small replication errors accumulate over time, along with radiation damage and other mutations or alterations of the genomic sequence. Instead of evolutionary pressure selecting organisms based on morphological traits, selection occurs at the level of mutations. Consequently, the only mutations observed in genes from healthy organisms are those that don't result in the organisms' death.

Because these changes between gene sequences are incremental, we can take homologous genes—genes with common evolutionary origin and related function—from a number of divergent organisms and compare them by aligning identical or similar residues. This comparison of multiple sequences shows which regions of a gene (or its derived protein) are sensitive to mutation and which are tolerant of having one residue replaced by another. Thus, we can develop hypotheses about the molecular events underlying the evolution of these sequences. Many bioinformatics methods, including pairwise sequence comparison and sequence database searching, are based on this observation that homologous genes have similar sequences.

In considering sequence similarity, there is one additional wrinkle to bear in mind: the difference between orthologs and paralogs. The chemical processes of molecular evolution are responsible for more than just giving rise to species differences. Evolutionary change can occur within the genome of a single species as well. *Orthologs* are genes that are evolutionarily related, share a function, and have diverged by speciation. *Paralogs*, on the other hand, have a common ancestor but have diverged by gene duplication and no longer have a common functional role. In other words, orthologs have the same function but occur in different species, while paralogs exist in the same genome but have different functions. A sequence database search may return both orthologs and paralogs. Depending on the objectives of your study, you probably will not want to treat them all as members of the same set.

8.2 Multiple Sequence Alignment

Multiple sequence alignment techniques are most commonly applied to protein sequences; ideally they are a statement of both evolutionary and structural similarity among the proteins encoded by each sequence in the alignment. We know that proteins with closely related functions are similar in both sequence and structure from organism to organism, and that sequence tends to change more rapidly than structure in the course of evolution. In multiple alignments generated from sequence data alone, regions that are similar in sequence are usually found to be superimposable in structure as well.

With a detailed knowledge of the biochemistry of a protein, you can create a multiple alignment by hand. This is a painstaking process, however. The challenge of automatic alignment is that it is hard to define exactly what an optimal multiple alignment is, and impossible to set a standard for a single correct multiple alignment. In theory, there is one underlying evolutionary process and one evolutionarily correct alignment to be generated from any group of sequences. However, the differences between sequences can be so great in parts of an alignment that there isn't an apparent, unique solution to be found by an alignment algorithm. Those same divergent regions are often structurally unalignable as well. Most of the insight that we derive from multiple alignments comes from analyzing the regions of similarity, not from attempting to align the very diverged regions.

The dynamic programming algorithm used for pairwise sequence alignment can theoretically be extended to any number of sequences. However, the time and memory requirements of this algorithm increase exponentially with the number of sequences. Dynamic programming alignment of two sequences takes seconds. Alignment of four relatively short sequences takes a few hours. Beyond that, it becomes impractical to align sequences this way. The program MSA is an implementation of an algorithm that reduces the complexity of the dynamic programming problem for multiple sequences to some extent. It can align about seven relatively short (200 -300) protein sequences in a reasonable amount of time. However, MSA is of little use when comparing large numbers of sequences.

8.2.1 Progressive Strategies for Multiple Alignment

A common approach to multiple sequence alignment is to progressively align pairs of sequences. The general progressive strategy can be outlined as follows: a starting pair of sequences is selected and aligned, then each subsequent sequence is aligned to the previous alignment. Like the Needleman-Wunsch and Smith-Waterman algorithms for sequence alignment, progressive alignment is an instance of a heuristic algorithm. Specifically, it is a greedy algorithm. *Greedy algorithms* decompose a problem into pieces, then choose the best solution to each piece without paying attention to the problem as a whole. In the case of progressive alignment, the overall problem (alignment of many sequences) is decomposed into a series of pairwise alignment steps.

Because it is a heuristic algorithm, progressive alignment isn't guaranteed to find the best possible alignment. In practice, however, it is efficient and produces biologically meaningful results. Progressive alignment methods differ in several respects: how they choose the initial pair of sequences to align, whether they align every subsequent sequence to a single cumulative alignment or create subfamilies, and how they score individual alignments and alignments of individual sequences to previous alignments.

8.2.2 Multiple Alignment with ClustalW

One commonly used program for progressive multiple sequence alignment is ClustalW. The heuristic used in ClustalW is based on phylogenetic analysis. First, a pairwise distance matrix for all the sequences to be aligned is generated, and a guide tree is created using the neighbor-joining algorithm. Then, each of the most closely related pairs of sequences—the outermost branches of the tree—are aligned to each other using dynamic programming. Next, each new alignment is analyzed to build a sequence profile. Finally, alignment profiles are aligned to each other or to other sequences (depending on the topology of the tree) until a full alignment is built.

This strategy produces reasonable alignments under a range of conditions. It's not foolproof; for distantly related sequences, it can build on the inaccuracies of pairwise alignment and phylogenetic analysis. But for sequence sets with some recognizably related pairs, it builds on the strengths of these methods. Pairwise sequence alignment by dynamic programming is very accurate for closely related sequences regardless of which scoring matrix or penalty values are used. Phylogenetic analysis is relatively unambiguous for closely related sequences. Using multiple sequences to create profiles increases the accuracy of pairwise alignment for more distantly related sequences.

There are many parameters involved in multiple sequence alignment. There are, of course, scoring matrices and gap penalties associated with the pairwise alignment steps. In addition, there are weighting parameters that alter the scoring matrix used in sequence-profile and profile-profile alignments. In ClustalW, these are set from the Multiple Alignment submenu or the Profile Structure Alignments submenu. In ClustalX, they are set from the Alignment pulldown menu.

The pairwise alignment parameters are familiar and have the same meaning in multiple alignment as they do in pairwise alignment. The multiple alignment parameters include gap opening and gap extension penalties for the multiple alignment process—to be used when fine-tuning alignments—and a maximum allowable delay, in terms of sequence length, for the start of divergent sequences at the beginning of the alignment.

One of ClustalW's heuristics is that, in protein sequence alignment, different scoring matrices are used for each alignment based on expected evolutionary distance. If two sequences are close neighbors in the tree, a scoring matrix optimized for close relationships aligns them. Distant neighbors are aligned using matrices optimized for distant relationships. Thus, when prompted to choose a series of matrices in the Multiple Alignment Parameters menu, it means just that: use BLOSUM62 for close relationships and BLOSUM45 for more distant relationships, rather than the same scoring matrix for all pairwise alignments.

Another heuristic that ClustalW uses is scalable gap penalties for protein profile alignments. A gap opening next to a conserved hydrophobic residue can be penalized more heavily than a gap opening next to a hydrophilic residue. A gap opening too close to another gap can be penalized more heavily than an isolated gap. These parameters are set in the Protein Gap Parameters menu.

Although ClustalW is run from the Unix command line, it is menu-driven and doesn't rely on command-line options. To start the program, you can simply type *clustalw*, and a menu of options is presented:

```
**************************************************************
******** CLUSTAL W (1.8) Multiple Sequence Alignments  ********
**************************************************************
     1. Sequence Input From Disc
     2. Multiple Alignments
     3. Profile / Structure Alignments
     4. Phylogenetic trees

     S. Execute a system command
     H. HELP
     X. EXIT (leave program)
```

This menu, along with subsequent menus that appear after you input your sequences, guides you through the use of ClustalW in a fairly straightforward fashion. Alignments are written in a plain-text format.

While ClustalW is simple to install and use on a Linux workstation, ClustalX, the X Windows-based graphical user interface for ClustalW, isn't so easy to compile. However, ClustalX runs in its own X window, has pulldown menus, and allows viewing and plotting of multiple sequence alignments in a color-coded format. It also allows you to append sequences to an alignment one at a time, and to produce color PostScript output of specified sequence ranges in an alignment from different files, if desired, along with other convenient features. To install ClustalX on a Linux machine, you need:

- The ClustalX binaries
- The NCBI software toolkit source distribution
- The LessTif libraries

The first thing you need to do is install the LessTif libraries. This distribution is extremely easy to work with. The LessTif libraries are available from http://www.lesstif.org and may also be available within your Linux distribution. The NCBI Toolkit (available from http://www.ncbi.nlm.nih.gov) should compile completely as long as your LessTif libraries are installed in */usr/X11R6/lib*. If the NCBI Toolkit installation produces the file *libvibrant.a*, the command *clustalx* will work.

8.2.3 Viewing and Editing Alignments with Jalview

can download if you're feeling adventurous. The other way to compile the software is to use the C versions of the programs that are now available. Because these programs have been automatically translated from Pascal, they require that the p2c (Pascal-to-C) libraries are installed on your system.

An easier approach for the novice is to use the sequence logo web server at the University of Cambridge, which (as of this writing) is actually recommended by the author of the DELILA programs and hence, we assume, does exactly what it's supposed to do. Aligned sequences can be submitted to this server in FASTA alignment format, which can be generated by ClustalX.

8.3 Phylogenetic Analysis

Having covered some of the basics of multiple sequence alignment, we now introduce one of its applications: phylogenetic inference. Phylogenetic inference is the process of developing hypotheses about the evolutionary relatedness of organisms based on their observable characteristics. Traditionally, phylogenetic analyses have been based on the gross anatomy of species. When Linneaus developed the system of classification into kingdoms, phyla, genera, and species, the early biologists sorted living things into a symbolic Tree of Life, which we saw in Figure 1-3. This tree-based representation of the relationships among species is a phylogenetic tree; it has since been adopted as a convenient schematic for depicting evolutionary relatedness based on sequence similarity. The quantitative nature of sequence relationships has allowed the development of more rigorous methods and rules for tree drawing.

While hand-drawn trees of life may branch fancifully according to what is essentially an artist's conception of evolutionary relationships, modern phylogenetic trees are strictly binary; that is, at any branch point, a parent branch splits into only two daughter branches. Binary trees can approximate any other branching pattern, and the assumption that trees are binary greatly simplifies the tree-building algorithms.

The length of branches in a quantitative phylogenetic tree can be determined in more than one way. Evolutionary distance between pairs of sequences, relative to other sequences in an input data set, is one way to assign branch length.

While a phylogeny of species generally has a root, assuming that all species have a specific common ancestor, a phylogenetic tree derived from sequence data may be rooted or unrooted. It isn't too difficult to calculate the similarity between any two sequences in a group and to determine where branching points belong. It is much harder to pinpoint which sequence in such a tree is the common ancestor, or which pair of sequences can be selected as the first daughters of a common ancestor. While some phylogenetic inference programs do offer a hypothesis about the root of a tree, most simply produce unrooted trees. Figure 8-2 and Figure 8-3 illustrate rooted and unrooted phylogenetic trees.

Figure 8-2. A rooted phylogenetic tree

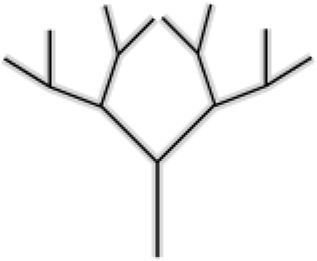
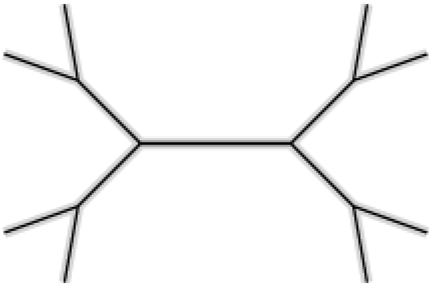


Figure 8-3. An unrooted phylogenetic tree



A phylogeny inferred from a protein or nucleic acid sequence has only a passing resemblance to a whole-organism tree of life (a true tree) that represents actual speciation events. A single phylogeny may be a tree, and it may describe a biological entity, but it takes far more than a single evolutionary analysis to draw conclusions about whole-organism phylogeny. Sequence-based phylogenies are quantitative. When they are built based on sufficient amounts of data, they can provide valuable, scientifically valid evidence to support theories of evolutionary history. However, a single sequence-based phylogenetic analysis can only quantitatively describe the input data set. It isn't valid as a quantitative tool beyond the bounds of that data set, and if you are using phylogenetic analysis tools to develop evolutionary hypotheses, it is critical to remember this point.

It has been shown, by comparative analysis of phylogenies generated for different protein and gene families, that one protein may evolve more quickly than another, and that a single protein may evolve more quickly in some organisms than in others. Thus, the phylogenetic analysis of a sequence family is most informative about the evolution of that particular gene. Only by analysis of much larger sets of data can theories of whole-organism phylogeny be suggested.

8.3.1 Phylogenetic Trees Based on Pairwise Distances

One of the easiest to understand algorithms for tree drawing is the pairwise distance method. This method produces a rooted tree. The algorithm is initialized by defining a matrix of distances between each pair of sequences in the input set. Sequences are then clustered according to distance, in effect building the tree from the branches down to the root.

Distances can be defined by more than one measure, but one of the more common and simple measures of dissimilarity between DNA sequences is the Jukes-Cantor distance, which is logarithmically related to the fraction of sites at which two sequences in an alignment differ. The fraction of matching positions in an ungapped alignment between two unrelated DNA sequences approaches 25%. Consequently, the Jukes-Cantor distance is scaled such that it approaches infinity as the fraction of unmatched residue pairs approaches 75%.

The pairwise clustering procedure used for tree drawing (UPGMA, unweighted pair group method using arithmetic averages) is intuitive. To begin with, each sequence is assigned to its own cluster, and a branch (or leaf) of the tree is started for that sequence at height zero in the tree. Then, the two clusters that are closest together in terms of whatever distance measure has been chosen are merged into a single cluster. A branch point (or node) is defined that connects the two branches. The node is placed at a height in the tree that reflects the distance between the two leaves that have been joined. This process is repeated iteratively, until there are only two clusters left. When they are joined, the root of the tree is defined. The branch lengths in a tree constructed using this process theoretically reflect evolutionary time.

8.3.2 Phylogenetic Trees Based on Neighbor Joining

Neighbor joining is another distance matrix method. It eliminates a possible error that can occur when the UPGMA method is used. UPGMA produces trees in which the branches that are closest together by absolute distance are placed as neighbors in the tree. This assumption places a restriction on the topology of the tree that can lead to incorrect tree construction under some conditions.

In order to get around this problem, the neighbor-joining algorithm searches not just for minimum pairwise distances according to the distance metric, but for sets of neighbors that minimize the total length of the tree. Neighbor joining is the most widely used of the distance-based methods and can produce reasonable trees, especially when evolutionary distances are short.

8.3.3 Phylogenetic Trees Based on Maximum Parsimony

A more widely used algorithm for tree drawing is called parsimony. *Parsimony* is related to Occam's Razor, a principle formulated by the medieval philosopher William of Ockham that states the simplest explanation is probably the correct one.[3] Parsimony searches among the set of possible trees to find the one requiring the least number of nucleic acid or amino acid substitutions to explain the observed differences between sequences.

[3] Or, in other words, "It is futile to do with more what can be done with fewer."

The only sites considered in a parsimony analysis of aligned sequences are those that provide evolutionary information—that is, those sites that favor the choice of one tree topology over another. A site is considered to be informative if there is more than one kind of residue at the site, and if each type of residue is represented in more than one sequence in the alignment. Then, for each possible tree topology, the number of inferred evolutionary changes at each site is calculated. The topology that is maximally parsimonious is that for which the total number of inferred changes at all the informative sites is minimized. In some cases there may be multiple tree topologies that are equally parsimonious.

As the number of sequences increases, so does the number of possible tree topologies. After a certain point, it is impossible to exhaustively enumerate the scores of each topology. A shortcut algorithm that finds the maximally parsimonious tree in such cases is the branch-and-bound algorithm. This algorithm establishes an upper bound for the number of allowed evolutionary changes by computing a tree using a fast or arbitrary method. As it evaluates other trees, it throws out any exceeding this upper bound before the calculation is completed.

8.3.4 Phylogenetic Trees Based on Maximum Likelihood Estimation

Maximum likelihood methods also evaluate every possible tree topology given a starting set of sequences. Maximum likelihood methods are probabilistic; that is, they search for the optimal choice by assigning probabilities to every possible evolutionary change at informative sites, and by maximizing the total probability of the tree. Maximum likelihood methods use information about amino acid or nucleotide substitution rates, analogous to the substitution matrices that are used in multiple sequence alignment.

8.3.5 Software for Phylogenetic Analysis

There is a variety of phylogenetic analysis software available for many operating systems. With such a range of choices, which package do you use? One of the most extensive listings currently available is maintained by Dr. Joe Felsenstein, the author of the PHYLIP package, and is accessible from the PHYLIP web page (http://evolution.genetics.washington.edu/phylip.html). If you don't want to follow our example and use PHYLIP, you can easily find information about other packages.

8.3.5.1 PHYLIP

The most widely distributed phylogenetic analysis package is PHYLIP. It contains 30 programs that implement different phylogenetic analysis algorithms. Each of the programs runs separately, from the command line. By default, most of the programs look for an input file called *infile* and write an output file called *outfile*. Rather than entering parameters via command-line flags, as with BLAST, the programs have an interactive text interface that prompts you for information.

The following are the PHYLIP programs you are most likely to use when you're just getting started analyzing protein and DNA sequence data:

PROTPARS

> Infers phylogenies from protein sequence input using the parsimony method

PROTDIST

> Computes an evolutionary distance matrix from protein sequence input, using maximum likelihood estimation

DNAPARS

Infers phylogenies from DNA sequence input using parsimony

DNAPENNY

Finds all maximally parsimonious phylogenies for a set of sequences using a branch-and-bound search

DNAML

Infers phylogenies from DNA sequence input using maximum likelihood estimation

DNADIST

Computes a distance matrix from DNA sequence input using the Jukes-Cantor distance or one of three other distance criteria

NEIGHBOR

Infers phylogenies from distance matrix data using either the pairwise clustering or the neighbor joining algorithm

DRAWGRAM

Draws a rooted tree based on output from one of the phylogeny inference programs

DRAWTREE

Draws an unrooted tree based on output from one of the phylogeny inference programs

CONSENSE

Computes a consensus tree from a group of phylogenies

RETREE

Allows interactive manipulation of a tree by the user—not based on data

PHYLIP is a flexible package, and the programs can be used together in many ways. To analyze a set of protein sequences with PHYLIP, you can:

1. Read a multiple protein sequence alignment using PROTDIST and create a distance matrix.
2. Input the distance matrix to NEIGHBOR and generate a phylogeny based on neighbor joining.
3. Read the phylogeny into DRAWTREE and produce an unrooted phylogenetic tree.

Or, you can:

1. Read a multiple sequence alignment using PROTPARS and produce a phylogeny based on parsimony.
2. Read the phylogeny using DRAWGRAM and produce a rooted tree.

Each of the PHYLIP programs is exhaustively documented in the **.doc* files available with the PHYLIP distribution. This documentation has been converted into HTML by several groups. Links to web-based PHYLIP documentation are available from the PHYLIP home page.

Several organizations have made PHYLIP servers available on the Web; the version of PHYLIP in the SDSC Biology Workbench produces downloadable PostScript output.

8.3.5.1.1 The PHYLIP input format

PHYLIP's molecular sequence analysis programs can accept sequence data in an aligned (interleaved) format:

```
39
Archaeopt  CGATGCTTAC CGCCGATGCT
Hesperorni CGTTACTCGT TGTCGTTACT
Baluchithe TAATGTTAAT TGTTAATGTT
B. virgini TAATGTTCGT TGTTAATGTT
Brontosaur CAAAACCCAT CATCAAAACC
B.subtilis GGCAGCCAAT CACGGCAGCC

TACCGCCGAT GCTTACCGC
CGTTGTCGTT ACTCGTTGT
AATTGTTAAT GTTAATTGT
CGTTGTTAAT GTTCGTTGT
CATCATCAAA ACCCATCAT
AATCACGGCA GCCAATCAC
```

where the first 10 characters are that sequence's name followed by the sequence in aligned form. Subsequent rows follow. In a sequential format, the complete first sequence is given, then the second complete sequence, etc. However, in either case, the sequences must be prealigned by another program. PHYLIP doesn't have a utility for computing multiple sequence alignments.

If you examine the phylogeny output from PHYLIP, you'll find it's represented by codes indicating each of the sequences, arranged in nested parentheses. This is called Newick notation. The pattern of the parentheses indicates the topology of the tree. The innermost parentheses surround the terminal branches of the tree, e.g., *(A,B)*, and each subsequent set of parentheses joins another pair of branches, e.g., *((A,B*),(*C,D))*. If the algorithm that generates the phylogeny produces branch lengths, these branch lengths are associated explicitly with each branch within the Newick notation: e.g., *((A:1.2,B:1.5):1.0,(C:2.5,D:0.5):1.2)*.

8.3.5.2 Generating input for PHYLIP with ClustalX

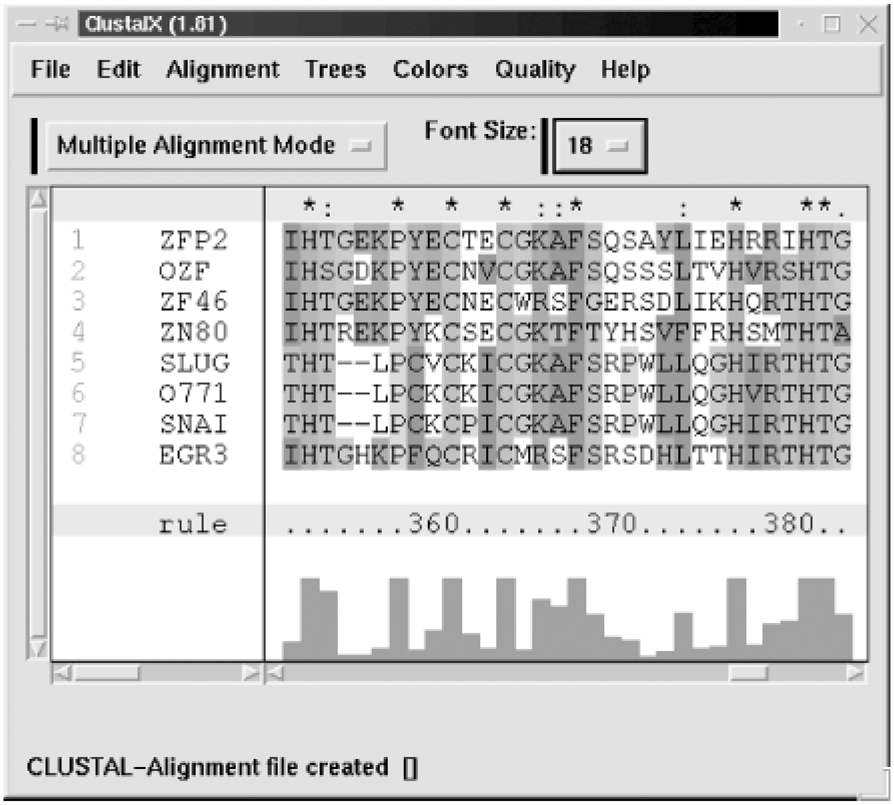
The multiple sequence alignment program ClustalX, which we discussed earlier in this chapter, draws phylogenetic trees with the neighbor joining method. Perhaps more importantly, it can read sequences in various input formats and then write PHYLIP-format files from multiple sequence alignments, using a simple Save As command from within the ClustalX interface.

8.4 Profiles and Motifs

In addition to studying relationships between sequences, one of the most successful applications of multiple sequence alignments is in discovering novel, related sequences. This profile- or motif-based analysis uses knowledge derived from multiple alignments to construct and search for sequence patterns. In this section, we first introduce some of the concepts behind motifs, then describe tools that use these principles to search sequence databases.

First, by way of a refresher, a multiple sequence alignment is an alignment of anywhere from three to hundreds of sequences. Multiple sequence alignments can span the full sequence of the proteins involved or a single region of similarity, depending on their purpose. Multiple sequence alignments, such as the one shown in Figure 8-4, are generally built up by iterative pairwise comparison of sequences and sequence groups, rather than by explicit multiple alignment.

Figure 8-4. A multiple sequence alignment, shown using ClustalX



```
                    *:   *  *  * ::*     :  *   **.
1    ZFP2     IHTGEKPYECTECGKAFSQSAYLIEHRRIHTG
2    OZF      IHSGDKPYECNVCGKAFSQSSSLTVHVRSHTG
3    ZF46     IHTGEKPYECNECWRSFGERSDLIKHQRTHTG
4    ZN80     IHTREKPYKCSECGKTFTYHSVFFRHSMTHTA
5    SLUG     THT--LPCVCKICGKAFSRPWLLQGHIRTHTG
6    O771     THT--LPCKCKICGKAFSRPWLLQGHVRTHTG
7    SNAI     THT--LPCKCPICGKAFSRPWLLQGHIRTHTG
8    EGR3     IHTGHKPFQCRICMRSFSRSDHLTTHIRTHTG

     rule     .......360.......370.......380..
```

CLUSTAL-Alignment file created []

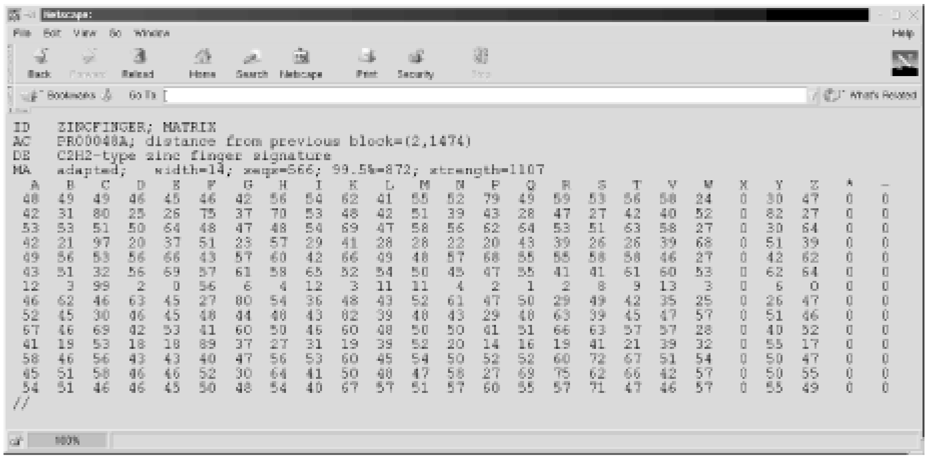
A sequence *motif* is a locally conserved region of a sequence, or a short sequence pattern shared by a set of sequences. The term "motif" most often refers to any sequence pattern that is predictive of a molecule's function, a structural feature, or family membership. Motifs can be detected in protein, DNA, and RNA sequences, but the most common use of motif-based analyses is the detection of sequence motifs that correspond to structural or functional features in proteins. Motifs are generated from multiple sequence alignments and can be displayed as patterns of amino acids (such as those in the Prosite database) or as sequence logos. For computational purposes, they can be represented as flexible patterns, position-specific scoring matrices, or profile hidden Markov models.

Motifs can be created for *protein families*, or sets of proteins whose members are evolutionarily related. Protein families can consist of many proteins that range from very similar to quite diverse. While the idea of a protein family is a fairly common concept, the method of selecting a protein family and defining its limits depends on the researcher who defines it. As in pairwise sequence comparison, there is a lower bound beyond which homology can't easily be detected. Motif-based methods can push this lower bound by detecting particularly subtle sequence patterns and distant homologs.

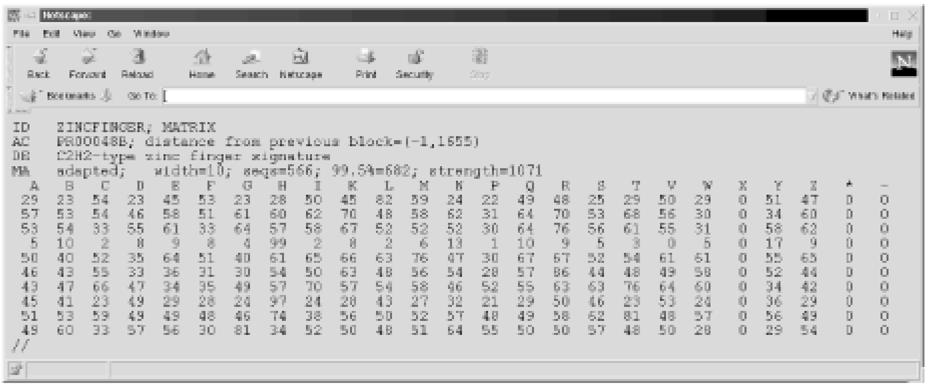
A sequence *profile* is a quantitative or qualitative method of describing a motif. A profile can be expressed in its most rudimentary form as a list of the amino acids occurring at each position in the motif. Early profile methods used simple profiles of this sort; however, modern profile methods usually weight amino acids according to their probability of being observed at each position.

Figure 8-5 shows a *position-specific scoring matrix* (PSSM), which is a matrix of scores representing a motif. Unlike a standard scoring matrix, the first dimension of the matrix is the length of the motif; the second dimension consists of the 20 amino acid possibilities. For each position in the matrix, there is a probability score for the occurrence of each amino acid. Most methods for developing position-specific scoring matrices normalize the raw probabilities with respect to a standard scoring matrix such as BLOSUM62.

Figure 8-5. PSSMs for sequence motifs common to zinc finger proteins


Netscape:
File Edit View Go Window Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go To:
What's Related
100%

```
ID   ZINCFINGER; MATRIX
AC   PR00048A; distance from previous block=(2,1474)
DE   C2H2-type zinc finger signature
MA   adapted;   width=14; seqs=566; 99.5%=872; strength=1107
   A   B   C   D   E   F   G   H   I   K   L   M   N   P   Q   R   S   T   V   W   X   Y   Z   *   -
  48  49  49  46  45  46  42  56  54  62  41  55  52  79  49  59  53  56  58  24   0  30  47   0   0
  42  31  80  25  26  75  37  70  53  48  42  51  39  43  28  47  27  42  40  52   0  82  27   0   0
  53  53  51  50  64  48  47  48  54  69  47  58  56  62  64  53  51  63  58  27   0  30  64   0   0
  42  21  97  20  37  51  23  57  29  41  28  28  22  20  43  39  26  26  39  68   0  51  39   0   0
  49  56  53  56  66  43  57  60  42  66  49  48  57  68  55  55  58  58  46  27   0  42  62   0   0
  43  51  32  56  69  57  61  58  65  52  54  50  45  47  55  41  41  61  60  53   0  62  64   0   0
  12   3  99   2   0  56   6   4  12   3  11  11   4   2   1   2   8   9  13   3   0   6   0   0   0
  46  62  46  63  45  27  80  54  36  48  43  52  61  47  50  29  49  42  35  25   0  26  47   0   0
  52  45  30  46  45  48  44  48  43  82  39  48  43  29  48  63  39  45  47  57   0  51  46   0   0
  67  46  69  42  53  41  60  50  46  60  48  50  50  41  51  66  63  57  57  28   0  40  52   0   0
  41  19  53  18  18  89  37  27  31  19  39  52  20  14  16  19  41  21  39  32   0  55  17   0   0
  58  46  56  43  43  40  47  56  53  60  45  54  50  52  52  60  72  67  51  54   0  50  47   0   0
  45  51  58  46  46  52  30  64  41  50  48  47  58  27  69  75  62  66  42  57   0  50  55   0   0
  54  51  46  46  45  50  48  54  40  67  57  51  57  60  55  57  71  47  46  57   0  55  49   0   0
//
```


Netscape:
File Edit View Go Window Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go To:
What's Related

```
ID   ZINCFINGER; MATRIX
AC   PR00048B; distance from previous block=(-1,1655)
DE   C2H2-type zinc finger signature
MA   adapted;   width=10; seqs=566; 99.5%=682; strength=1071
   A   B   C   D   E   F   G   H   I   K   L   M   N   P   Q   R   S   T   V   W   X   Y   Z   *   -
  29  23  54  23  45  53  23  28  50  45  82  59  24  22  49  48  25  29  50  29   0  51  47   0   0
  57  53  54  46  58  51  61  60  62  70  48  58  62  31  64  70  53  68  56  30   0  34  60   0   0
  53  54  33  55  61  33  64  57  58  67  52  52  52  30  64  76  56  61  55  31   0  58  62   0   0
   5  10   2   8   9   8   4  99   2   8   2   6  13   1  10   9   5   3   0   5   0  17   9   0   0
  50  40  52  35  64  51  40  61  65  66  63  76  47  30  67  67  52  54  61  61   0  55  65   0   0
  46  43  55  33  36  31  30  54  50  63  48  56  54  28  57  86  44  48  49  58   0  52  44   0   0
  43  47  66  47  34  35  49  57  70  57  54  58  46  52  55  63  63  76  64  60   0  34  42   0   0
  45  41  23  49  29  28  24  97  24  28  43  27  32  21  29  50  46  23  53  24   0  36  29   0   0
  51  53  59  49  49  48  46  74  38  56  50  52  57  48  49  58  62  81  48  57   0  56  49   0   0
  49  60  33  57  56  30  81  34  52  50  48  51  64  55  50  50  57  48  50  28   0  29  54   0   0
//
```

Finally, a *profile hidden Markov model* (HMM) is the rigorous probabilistic formulation of a sequence profile. Profile HMMs contain the same probability information found in a PSSM; however, they can also account for gaps in the alignment, which tends to improve their sensitivity. Because profile analysis methods are still a subject of active research, there are many different programs and methods for motif discovery and profile building. We will focus on two of the easiest motif discovery packages to use, MEME and HMMer. We also describe the searchable databases of preconstructed protein family motifs—some with associated PSSMs or profile HMMs—offered by several organizations.

8.4.1 Motif Databases

We have seen that profiles and other consensus representations of sequence families can be used to search sequence databases. It shouldn't be too surprising, then, that there are motif databases that can be searched using individual sequences. Motif databases contain representations of conserved sequences shared by a sequence family. Today, their primary use is in annotation of unknown sequences: if you get a new gene sequence hot off the sequencer, scanning it against a motif database is a good first indicator of the function of the protein it encodes.

Motifs are generated by a variety of methods and with different objectives in mind. Some rely on automated analysis, but there is often a large amount of hands-on labor invested in the database by an expert curator. Because they store only those motifs that are present in reasonably large families, motif databases are small relative to GenBank, and they don't reflect the breadth of the protein structure or sequence databases. Be aware that an unsuccessful search against a motif database doesn't mean your sequence contains no detectable pattern; it could be part of a family that has not yet been curated or that doesn't meet the criteria of the particular pattern database you've searched. For proteins that do match defined families, a search against the pattern databases can yield a lot of homology information very quickly.

8.4.1.1 Blocks

Blocks, a service of the Fred Hutchinson Cancer Research Center, is an automatically generated database of ungapped multiple sequence alignments that correspond to the most conserved regions of proteins. Blocks is created using a combination of motif-detection methods, beginning with a step that exhaustively searches all spaced amino acid triplets in the sequence to discover a seed alignment, followed by a step that extends the alignment to find an aligned region of maximum length. The Blocks database itself contains more than 4,000 entries; it is extended to over 10,000 entries by inclusion of blocks created from entries in several other protein family databases (see Section 8.4.1.6). The Blocks server also provides several useful search services, including IMPALA (which uses the BLAST statistical model to compare a sequence against a library of profiles) and LAMA (Local Alignment of Multiple Alignments; Shmuel Pietrokovski's program for comparing an alignment of your own sequences against a database of Blocks).

8.4.1.2 PROSITE

PROSITE is an expert-curated database of patterns hosted by the Swiss Institute of Bioinformatics. It currently contains approximately 1,200 records, and is available for download as a structured flat file from http://ftp.expasy.ch. PROSITE uses a single consensus pattern to characterize each family of sequences. Patterns in PROSITE aren't developed based on automated analysis. Instead, they are carefully selected based on data published in the primary literature or on reviews describing the functionality of specific groups of proteins. A humorous cartoon on the PROSITE server indicates that the optimal method for identifying patterns requires only a human, chalk, and a chalkboard. PROSITE contains pattern information as well as position-specific scoring matrices that can detect new instances of the pattern.

8.4.1.3 Pfam

Pfam is a database of alignments of protein domain families. Pfam is made up of two databases: Pfam-A and Pfam-B. Pfam-A is a curated database of over 2,700 gapped profiles, most of which cover whole protein domains; Pfam-B entries are generated automatically by applying a clustering method to the sequences left over from the creation of Pfam-A. Pfam-A entries begin with a *seed alignment*, a multiple sequence alignment that the curators are confident is biologically meaningful and that may involve some manual editing. From each seed alignment, a profile hidden Markov model is constructed and used to search a nonredundant database of available protein sequences. A full alignment of the family is produced from the seed alignments and any new matches. This process can be iterated to produce more extensive families and detect remote matches. Pfam entries are annotated with information extracted from the scientific literature, and incorporate structural data where available. As a final note, Pfam is the database of profile HMMs used by the GeneWise genefinder to search for open reading frames.

8.4.1.4 PRINTS

PRINTS is a database of protein motifs similar to PROSITE, except that it uses "fingerprints" composed of more than one pattern to characterize an entire protein sequence. Motifs are often short relative to an entire protein sequence. In PRINTS, groups of motifs found in a sequence family can define a signature for that family.

8.4.1.5 COG

NCBI's Clusters of Orthologous Groups (COG) database is a different type of pattern database. COG is constructed by comparing all the protein sequences encoded in 21 complete genomes. Each cluster must consist of protein sequences from at least three separate genomes. The premise of COG is that proteins that are conserved across these genomes from many diverse organisms represent ancient functions that have been conserved throughout evolution. COG entries can be accessed by organism or by functional category from the NCBI web site. COG currently contains more than 2,100 entries.

8.4.1.6 Accessing multiple databases

So, which motif database should you use to analyze a new sequence? Because the comparisons are performed quickly and efficiently, we recommend you use as many as possible, keeping track of the best matches from each, their scores, and (if available) the significance of the hit. While Blocks uses InterPro as one of the sources for its own patterns, as of June 2000 it contains only ungapped patterns, omitting gapped profiles such as those contained in Pfam-A and PROSITE. Fortunately, all the motif databases discussed here have search interfaces available on the Web, most of which accept input in FASTA format or FASTA alignment format.

One service that allows integrated searching of many motif databases is the European Bioinformatics Institute's Integrated Resource of Protein Domains and Functional Sites (InterPro to its friends). InterPro allows you to compare a sequence against all the motifs from Pfam, PRINTS, ProDom, and PROSITE. InterPro motifs are annotated with the name of the source protein, examples of proteins in which the motif occurs, references to the literature, and related motifs.

8.4.2 Constructing and Using Your Own Profiles

Motif databases are useful if you're looking for protein families that are already well documented. However, if you think you've found a new motif you want to use to search GenBank, or you want to get creative and look for patterns in unusual places, you need to build your own